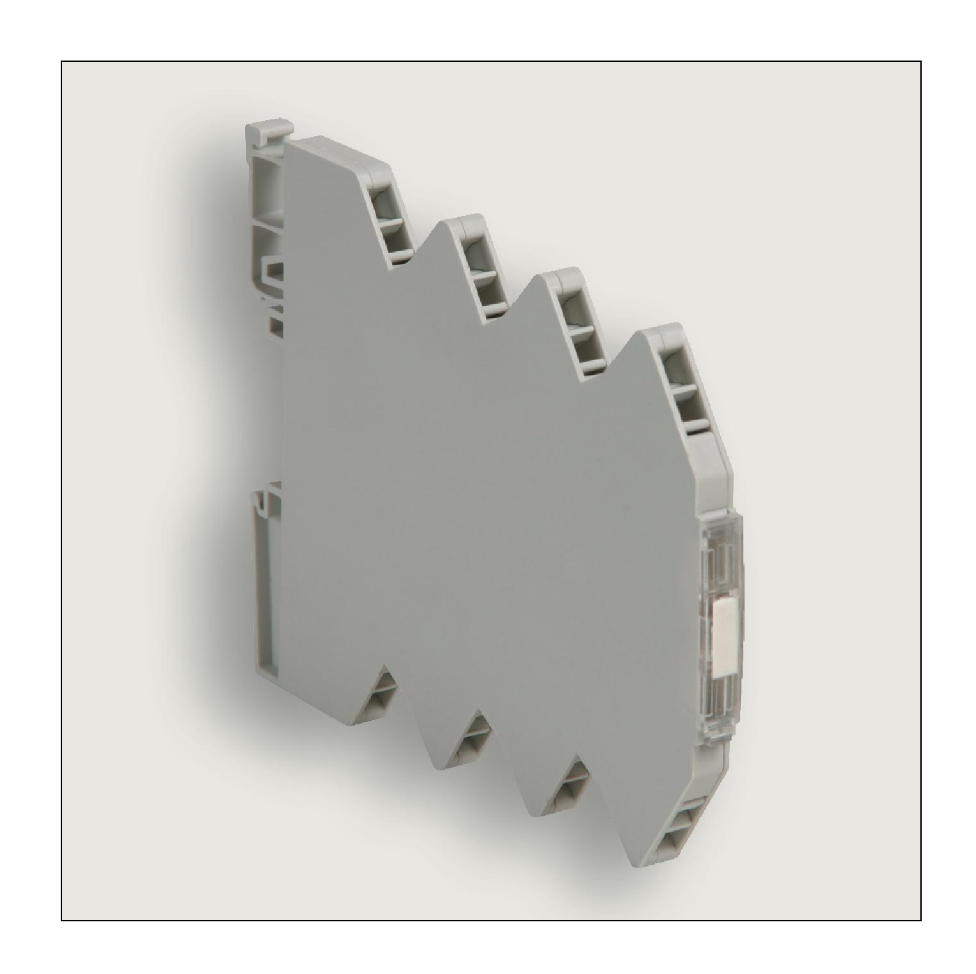


Automation Solutions

Handbuch

CANopen – Gateway 716459

Beschreibung von CANopen – Gateway in Verbindung mit LOCC-Box-Net 716410.

Version 1.00



Das vorliegende Handbuch ist Bestandteil des Gerätes und enthält wichtige Informationen zur Sicherheit und Bedienung. Lesen Sie vor dem Gebrauch das Handbuch, um mögliche Gefahren auszuschließen und den einwandfreien Gebrauch zu gewährleisten.

Dieses Handbuch und die darin enthaltenen Informationen wurden mit der gebotenen Sorgfalt zusammengestellt. Die Firma Lütze übernimmt jedoch keine Gewähr für Druck- oder andere Fehler oder daraus entstehende Schäden.

So können Sie uns erreichen

Friedrich Lütze GmbH & Co. KG
Postfach 1224
D-71366 Weinstadt - Großheppach
Germany

Telefon - Zentrale:	+49/ (0)7151/ 6053-0
Telefax:	+49/ (0)7151/ 6053-277
E-Mail:	automation@luetze.de
Internet:	http://www.luetze.com



Inhaltsverzeichnis


LUTZE

1 Allgemeine Hinweise

1.1 Symbolerklärung

Die Betriebsanleitung enthält Sicherheitshinweise die durch ein Signalwort in Kombination mit einer bestimmten Farbe. Die Kombination steht für die jeweilige Warnstufe. Die Hinweise weisen möglich Gefahren auf und geben Hinweise zur Vermeidung.

	Weist auf eine gefährliche Situation hin, die bei Nichtbeachtung zum Tod oder schweren Verletzungen führt.
	Weist auf eine gefährliche Situation hin, die bei Nichtbeachtung zum Tod oder schweren Verletzungen führen kann.
	Weist auf eine gefährliche Situation hin, die bei Nichtbeachtung zu leichten oder mittleren Verletzungen führen kann.
	Weist auf eine Situation hin, die das Produkt oder die Umgebung schädigen könnte. Dieser Hinweis sieht von Verletzungen ab.

1.2 Urheberrecht

Das Handbuch ist nur für den Betreiber und dessen Personal bestimmt. Die Inhalte dürfen weder vollständig noch teilweise an Dritte weitergegeben, vervielfältigt, verwertet oder anderweitig mitgeteilt werden, soweit dies nicht ausdrücklich und schriftlich von der Firma Friedrich Lütze GmbH & Co. KG gestattet wurde. Inhaltliche Angaben, Texte, Bilder und Zeichnungen sind urheberrechtlich geschützt und unterliegen dem gewerblichen Schutzrecht. Zuwiderhandlungen können strafrechtliche Folgen nach sich ziehen. Die genannten Marken und Produktnamen in diesem Dokument sind Warenzeichen oder eingetragene Warenzeichen der Titelhalter.

1.3 Haftungsausschluss

Wir haben den Inhalt dieser Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, und notwendige Korrekturen sind in den nachfolgenden Ausgaben enthalten.
Für Verbesserungsvorschläge sind wir dankbar.
Für Schäden, die durch fehlende oder unzureichende Kenntnisse des Handbuches entstehen, ist jegliche Haftung durch die Fa. Friedrich Lütze GmbH & Co. KG ausgeschlossen. Für den Betreiber ist es deshalb ratsam, sich die Einweisung des Personals schriftlich bestätigen zu lassen.
Umbauten oder funktionelle Veränderungen an den Modulen sind aus Sicherheitsgründen nicht gestattet. Nicht ausdrücklich vom Hersteller genehmigte Umbauten an den Modulen führen deshalb zum Verlust jeglicher Haftungsansprüche gegen die Fa. Friedrich Lütze GmbH & Co. KG.
Das gilt ebenfalls, wenn nicht originale bzw. nicht von uns zugelassene Teile oder Ausstattungen verwendet werden.

1.4 Sicherheitshinweise

1.4.1 Inhalt des Handbuches

Das Handbuch ist vor allen Arbeiten, die am oder mit dem Gerät ausgeführt werden zu lesen und dementsprechend einzuhalten. Dies gilt für alle Personen die mit dem Gerät in Kontakt kommen. Auch geschultes Personal und Fachkräfte, insbesondere Elektrofachkräfte, die bereits mit ähnlichen Geräten gearbeitet haben, sollten die Anleitung gelesen und verstanden haben.

1.4.2 Bestimmungsgemäße Verwendung

Die Geräte sind ausschließlich für den industrielen Einsatz konzipiert. Die bestimmungsgemäße Verwendung umfasst das Vorgehen gemäß Handbuch. Die Geräte dürfen nur für die in den technischen Unterlagen vorgesehenen Fälle und nur in Verbindung mit den von uns empfohlenen bzw. zugelassenen Fremdgeräten und Fremdkomponenten verwendet werden. Der einwandfreie und sichere Betrieb des Produkts setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung und Montage sowie sorgfältige Bedienung und Instandhaltung voraus.

1.4.3 Bedienpersonal

Nur qualifiziertes Personal darf folgende Arbeiten an den Modulen durchführen:

- Installation
- Inbetriebnahme
- Betrieb
- Instandhaltung.

Qualifiziertes Personal im Sinne der sicherheitstechnischen Hinweise sind Personen, die die Berechtigung haben, Geräte, Systeme und Stromkreise gemäß den Standards der Sicherheitstechnik in Betrieb zu nehmen, zu Erden und zu kennzeichnen. Das Bedienpersonal ist entsprechend einzuweisen und zu schulen.

1.4.4 Wartung

Die Module selbst sind wartungsfrei. Daher sind für den laufenden Betrieb keine Inspektions- und Wartungsintervalle nötig.

1.4.5 Stilllegung und Entsorgung

Für die Stilllegung und Entsorgung der Module hat die betreibende Firma die für den Standort geltenden Umweltrichtlinien des jeweiligen Landes zu beachten.

2 Gateway – CANopen, 716459

Das LOCC-Box-Net Interface (Gateway) ist eine Elektronik-Baugruppe zum Verteilen und Umsetzen der Daten und Nachrichten der seriellen LOCC-Box-Net Schnittstelle (LOCCbus) auf 3 weitere Kommunikations-Schnittstellen USB, CANopen oder RS232.

2.1 Allgemein

2.1.1 Erläuterungen

Die serielle LOCC-Box-Schnittstelle (LOCCbus) ist eine 1-Draht Kommunikationsschnittstelle. Diese ist gemäß LIN-Spezifikation ausgeführt. Das Protokoll auf dieser Schnittstelle ist an das Multidrop-Protokoll angelehnt. Der Telegrammaufbau der LOCC-Box-Schnittstelle wird im Kapitel 4.3 (RS-232) und 4.4 (CANopen) detailliert beschrieben.

Das Gateway unterstütz folgende Schnittstellen:

- Full-Speed USB-Schnittstelle mit einer maximalen Bitrate von 12 MBit/s gemäß USB 2.0
- CAN-Schnittstelle gemäß ISO 11898-1, physikalischer Übertragungslayer ist das CAN-High-Speed-Layer gemäß ISO 11898-2
- serielle RS232 Schnittstelle

Die USB-Schnittstelle ist zum Anschluss an einen handelsüblichen PC oder ein Notebook geeignet. Die USB-Schnittstelle wird unter Windows XP[1)], Windows Vista[1)] oder Windows7 als serielle COM-Schnittstelle erkannt. Zusammen mit der Software LOCC-Pads dient sie als Interface zur Erstinbetriebnahme und Konfiguration der LOCC-Box-Net Baugruppen.

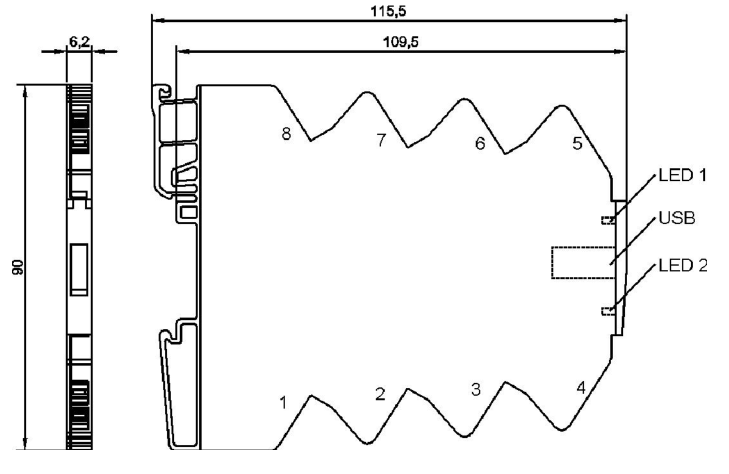
Die CANopen- wie auch die serielle RS232-Schnittstelle sind zum Anschluss an z.B. speicher-programmierbare Steuerungen (SPS) verschiedener Hersteller geeignet.

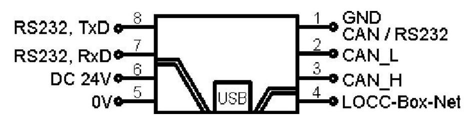
Ein gleichzeitiger Betrieb der USB- und CANopen-Schnittstelle ist nicht möglich. In diesem Fall hat die Kommunikation über die USB-Schnittstelle immer Vorrang. Die Kommunikation mit der CANopen-Schnittstelle ist dann abgeschaltet, das LOCC-Box-Net Interface ist dann als CANopen-Teilnehmer elektrisch auch nicht vorhanden (kein CAN-Acknowledge!).

Die CANopen-Schnittstelle arbeitet nach dem CANopen-Protokoll CiA DS301, als CANopen-Profil wird das „Generic IO Profil“ nach CiA DS401 verwendet. Die Baudrate und die CANopen-Knoten-Adresse (Node-ID) werden über die Software LOCC-Pads eingestellt.

Die Kommunikationsbeziehung der 3 Gateway-Schnittstellen (RS232, USB und CANopen) sind jeweils ausschließlich auf die Schnittstelle der LOCC-Box-Net Module gerichtet. Eine Quer-Kommunikation (USB <-> RS232 <-> CANopen) untereinander ist nicht vorgesehen.

2.1.2 Abmessungen und Anschlüsse:





1:	GND (Masse) - CANopen / RS232
2:	CANopen Anschluss - CAN_L
3:	CANopen Anschluss - CAN_H
4:	1 Draht Bus (LOCC-Box-Net)
5:	0V (Versorgung GW)
6:	DC 12/24V (+Versorgung GW)
7:	RS232 Anschluss - RxD
8:	RS232 Anschluss – TxD

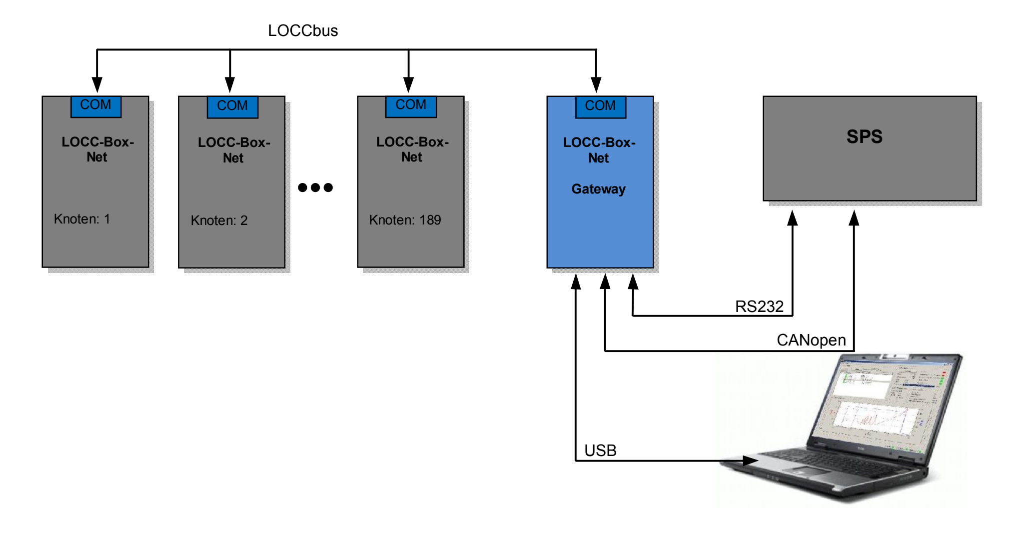
USB:	USB Anschluss
LED 1:	USB Status bei Datenaustausch über UART
LED 2:	CANopen Status

2.1.3 Funktion und Anzeigen

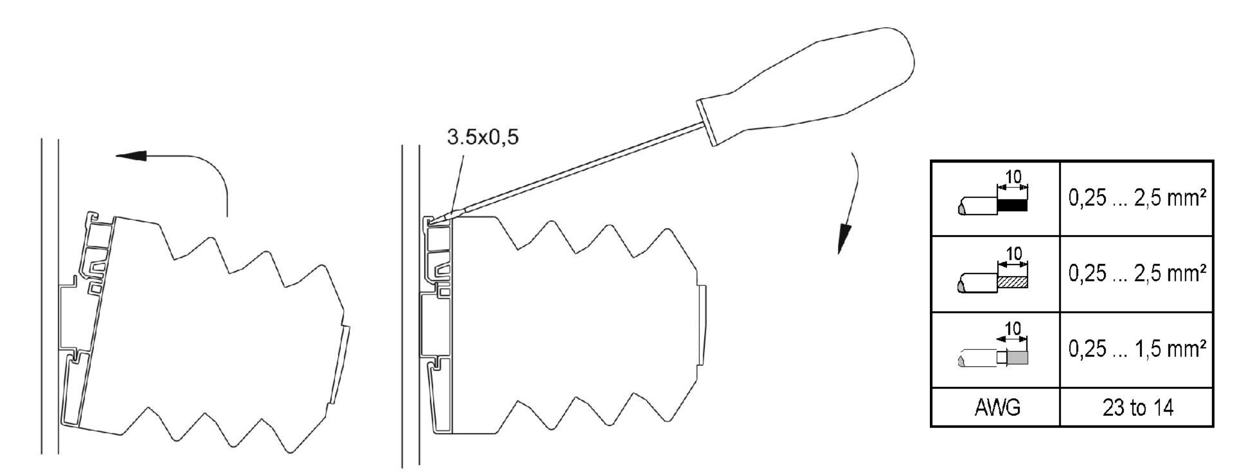
Funktion	PIN	Beschreibung
GND (Masse)	1	Bezugsmasse für CANopen und RS232, Anschluss für Schirm
CAN_L	2	Anschluss der CAN_Low Leitung, Abschlusswiderstand typisch 120R, muss extern gesetzt werden
CAN_H	3	Anschluss der CAN_High Leitung, Abschlusswiderstand typisch 120R, muss extern gesetzt werden
1 Draht Bus	4	1 Draht Kommunikationsbus zum Anschluss von typisch 40 und maximal 84 LOCC-Box-Net Modulen, siehe Topologie (unten). Die Buslänge beträgt max. 40m.
- Anschluss, Eigenversorgung	5	Ist direkt mit der Spannungsversorgung zu verbinden.
+ Anschluss, Eigenversorgung	6	Ist direkt mit der Spannungsversorgung zu verbinden.
RS232_RxD	7	RS232 Empfangsdatenleitung
RS232_TxD	8	RS232 Sendedatenleitung

Anzeigen	Funktion	Beschreibung
LED 1, grün - leuchtend	USB	USB ist aktiv
LED 1, grün - austastend	USB + RS232	USB ist aktiv. Erfolgt ein zusätzlicher Datenaustausch über die RS232 Schnittstelle, so wird die LED für diese Zeit ausgetastet.
LED 1, grün - aufblitzend	USB + RS232	USB ist nicht aktiv. Erfolgt ein Datenaustausch über die RS232 Schnittstelle, so blitzt die LED für diese Zeit auf.
LED 1, rot - leuchtend	Firmware	Die Firmware wurde gelöscht. Wenn die LED 2 ebenfalls rot leuchtet, so befindet sich das Gateway im Firmware-Update-Modus.
LED 2, grün - blinkend	CANopen	CANopen ist aktiv – Pre-operational Modus
LED 2, grün - aufblitzend	CANopen	CANopen ist aktiv - Stopp Modus
LED 2, grün - leuchtend	CANopen	CANopen ist aktiv - Betriebs Modus
LED 2, rot - aufblitzend	CANopen	CANopen ist aktiv - BUS_WARN Meldung
LED 2, rot - leuchtend	CANopen	CANopen ist aktiv - BUS_OFF Meldung, In Verbindung mit LED1 rot leuchtend befindet sich das Gateway im Firmware-Update-Modus.

2.1.4 Topologie und Aufbau

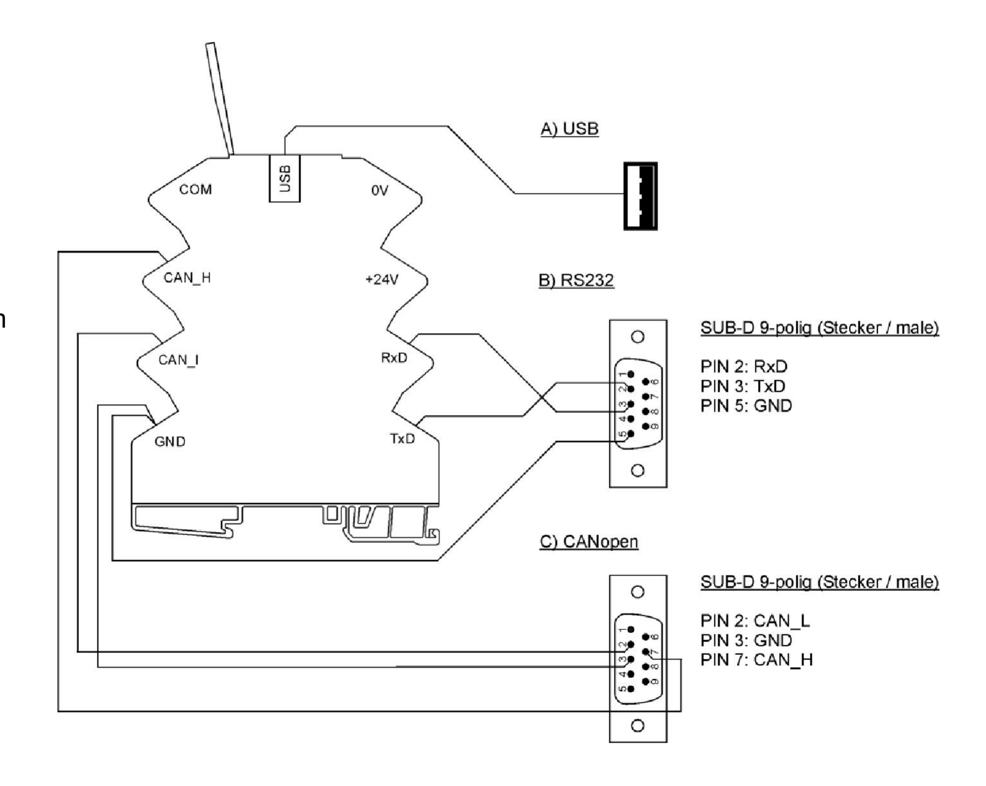


2.1.5 Montage



2.2 Installation

1. Versorgen Sie das Gateway und alle LOCC-Box-Net Module mit DC 12/24V Betriebsspannung.
2. Verbinden Sie alle "COM" Anschlüsse der Module mit dem Gateway.
   Hierzu eignen sich die im Zubehör angegebenen Brückungskämme. Siehe Abschnitt 7 Zubehör.
3. Schließen Sie das Gateway wie folgt an:
   A) USB
   B) RS232
   C) CANopen



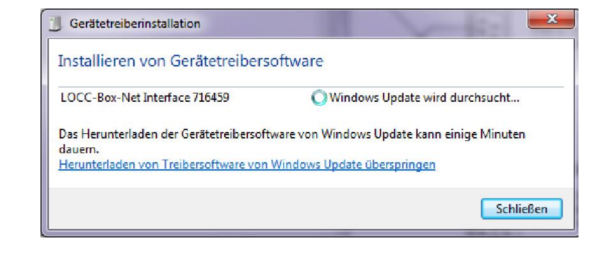
2.2.1 Anschluss an USB

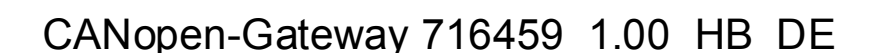


Installieren Sie die neueste Version von LOCC-Pads! Hierbei wird der Gerätetreiber in das Windows-Systemverzeichnis kopiert.

Verbinden Sie das Gateway mit dem PC / Laptop mit Hilfe des mitgelieferten USB-Kabels.
Beim erstmaligen Anschluss wird das Gateway als neue Hardware "LOCC-Box-Net Interface 716459" erkannt und automatisch installiert.

Für eine manuelle Installation lesen Sie bitte Kapital 4.5.





2.3 Kommunikation mit LOCC-Pads

Siehe Handbuch „LOCC-Box-Net_x.xx_HB_DE“

2.4 Kommunikation über RS232

Die UART-Schnittstelle RS232 ermöglicht eine Kommunikation zwischen Gateway (LOCC-Box-Net) und einem Terminal Programm, z.B. Hyperterminal von Microsoft, oder einer SPS. Die Telegramme beginnen mit einem <STX> (ASCII 0x02) und enden mit einem <ETX> (ASCII 0x03).
<STX> entspricht der Tastenkombination “Strg+B“ und <ETX> “Strg+C“. Die einzelnen Zahlen werden als ASCII-HEX übertragen. Bitte verwenden Sie folgende Einstellungen:

- Baudrate: 600 - 115200 bit/s
- Datenbit: 8
- Parität: keine
- Stoppbits: 1
- Flusssteuerung: kein

Die Kommunikation gestattet das Schreiben und Lesen von Werten und Zuständen. Der Umfang entspricht der Darstellung wie in der Software LOCC-Pads. Folgende Informationen können gelesen werden:

- Modultyp
- Modulzustand / Konfiguration:
  - Modul AUS, AN, Ausgelöst“ und EXT.AUS
  - Neue Module am BUS
  - Stromwarnung (I>90%), Abschaltung durch Überstrom
  - Unterspannung
  - Abschaltung durch Kurzschluss
  - Systemfehler
  - Schalterstellung vom Strombereich
  - Schalterstellung vom Charakteristik
- Eingangsspannung
- Ausgangsspannung
- Strom
- Kennlinieneinstellungen
- Softwareversion
- Seriennummer
- Zählerstände

Geschrieben werden kann der Betriebszustand AN / AUS.



Beispiel: HyperTerminal

1. Rufen Sie das Programm über **Start/ Programme/ Zubehör/ Kommunikation/ HyperTerminal** auf.

2. Geben Sie den Namen für die neue Verbindung ein, und weisen Sie ihr ein Symbol zu. Z.B. Lütze. Bestätigen Sie mit "OK" zum Fortfahren oder "Abrechen" zum Verlassen.



3. Bitte wählen Sie im Feld "Verbindung herstellen über" die COM Schnittstelle aus an der das Gateway angeschlossen ist.

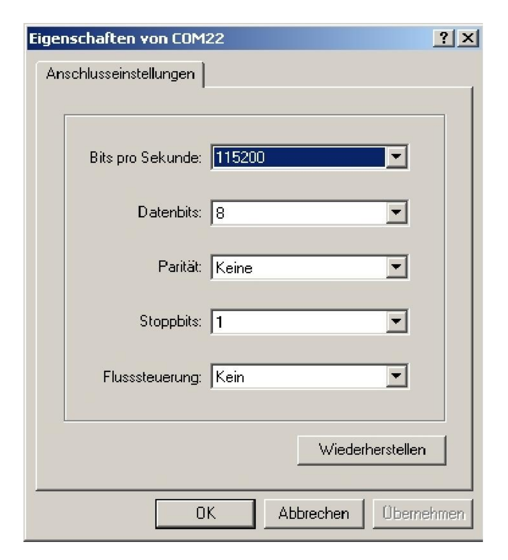


HINWEIS

Sollten Sie nicht wissen welcher Anschluss der richtige ist, so können Sie dies im Geräte-Manager nachsehen. Hierzu öffnen Sie über **Start/ Einstellungen/ Systemsteuerung** den Ordner **"System"**. Unter dem Reiter **"Hardware"** befindet sich der Geräte-Manager.

4. Tragen Sie hier die Eigenschaften für die Kommunikation ein. Verwenden Sie folgende Einstellungen:

   - Baudrate: 600 - 115200 bit/s
   - Datenbit: 8
   - Parität: keine
   - Stoppbits: 1
   - Flusssteuerung: kein
   -



HINWEIS

Die eingestellte Baudrate muss identisch mit der Einstellung im Gateway sein.
Siehe Kapitel 3.4.2 "Einstellung LOCC-Box Gateway für CANopen und RS 232".

5. Einstellung lokales Echo

   - Beenden Sie die Verbindung durch Anruf "Trennen".
   - Unter "Datei" - Eigenschaften öffnet sich das Menü "Eigenschaften von …", gehen Sie hier auf den Reiter Eigenschaften.
   - Wählen Sie den Button "ASII Konfiguration" und kreuzen Sie den Punkt "Eingehende Zeichen lokal ausgeben (lokales Echo)".



2.4.1 Werte und Zustände lesen

Abfrage	Kommando	Antwort
	***<STX>* = Strg+B / *<ETX>* = Strg+C**	
Softwareversion	<STX> nn 50 <ETX> Modul (Knoten) – Kommando	<STX> XX YY <ETX> YY = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB, LSB (Hex) = Version, versehen mit einer Kommastelle von links
	Beispiel: Eingabe: <STX>0150<ETX>, Abfrage Modul (Knoten) 1 Antwort: <STX>04 00<ETX> = 00 04 = 0.4	
Seriennummer	<STX> nn 51 <ETX> Modul (Knoten) – Kommando	<STX> WW XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) WW = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* = Seriennummer
	Beispiel: Eingabe: <STX>0151<ETX>, Abfrage Modul (Knoten) 1 Antwort: <STX>EF E1 01 00<ETX> = 00 01 E1 EF = *Dezimalwert* = 123375	
LOCC-Box Zähler Betriebsspg. An	<STX> nn 52 <ETX> Modul (Knoten) – Kommando	<STX> XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* = Zählerstand
	Beispiel: Eingabe: <STX>0B52<ETX>, Abfrage Modul (Knoten) 11 Antwort: <STX>0C 01 00<ETX> = 00 01 0C = *Dezimalwert* = 268	
LOCC-Box Zähler “Betriebsstunden (h)“	<STX> nn 53 <ETX> Modul (Knoten) – Kommando	<STX> XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* / 2 = Betriebsstunden (h)
	Beispiel: Eingabe: <STX>0C53<ETX>, Abfrage Modul (Knoten) 12 Antwort: <STX>60 01 00<ETX> = 00 01 60 = *Dezimalwert* / 2 = 176 h	
LOCC-Box Zähler “Betriebsstunden AN(h)“	<STX> nn 54 <ETX> Modul (Knoten) – Kommando	<STX> XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* / 2 = Betriebsstunden (h)
	Beispiel: Eingabe: <STX>0D54<ETX>, Abfrage Modul (Knoten) 13 Antwort: <STX>FB 08 00<ETX> = 00 08 FB = *Dezimalwert* / 2 = 1149,5 h	

Abfrage	Kommando	Antwort
	***<STX>* = Strg+B / *<ETX>* = Strg+C**	
LOCC-Box Zähler "Ausgelöst"	<STX> nn 55 <ETX> Modul (Knoten) – Kommando	<STX> XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* = Auslösung
	Beispiel: Eingabe: <STX>1155<ETX>, Abfrage Modul (Knoten) 17 Antwort: <STX>28 00 00<ETX> = 00 00 28 = *Dezimalwert* = 40	
LOCC-Box Zähler "Einschalten"	<STX> nn 56 <ETX> Modul (Knoten) – Kommando	<STX> XX YY ZZ <ETX> ZZ = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB … LSB (Hex) = Umwandlung in *Dezimalwert* = Auslösung
	Beispiel: Eingabe: <STX>0E56<ETX>, Abfrage Modul (Knoten) 14 Antwort: <STX>11 00 00<ETX> = 00 00 11 = *Dezimalwert* = 17	
Modulzustand/ Konfiguration	<STX> nn 04 <ETX> Modul (Knoten) – Kommando	<STX> X_2 X_1 Y_2 Y_1 <ETX> Y_1 = dezimal 0-9 = Strombereich 1-10A Y_2 = dezimal 0-9 = Charakteristik 1-10 X_1 (binär) 0 0 0 0 00: AUS (Taster / LOCC-Pads) 01: AN 10: Ausgelöst 11: EXT. AUS reserviert reserviert X_2 (binär) 0 0 0 0 1: Iwarnung (I > 0.9 * Inom) 1: Unterspannung (U < 10.0V) 1: Kurzschluss 1: Systemfehler
	Beispiel: Eingabe: <STX>0204<ETX>, Abfrage Modul (Knoten) 2 Antwort: <STX>4 2 9 1<ETX> 1 = dezimal 1 = Strombereich 2A 9 = dezimal 9 = Charakteristik 10 2: 0 0 1 0 10: Ausgelöst 0: kein neues Modul am BUS 4: 0 1 0 0 1: Kurzschluss	
Modultyp	<STX> nn 00 <ETX> Modul (Knoten) – Kommando	<STX> X_2X_1 <ETX> = dezimal = Typennummer
	Beispiel: Eingabe: <STX>0A00<ETX>, Abfrage Modul (Knoten) 10 Antwort: <STX>01<ETX> = dezimal = Typ 1	

Abfrage	**Kommando**	**Antwort**
	***<STX>* = Strg+B / *<ETX>* = Strg+C**	
Eingangsspg.	<STX>nn 0503<ETX> Modul (Knoten) — Kommando	<STX> XX YY <ETX> YY = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB, LSB (Hex) = Umwandlung in *Dezimalwert* Der maximale Messwert ist 1024 und entspricht 39V. Hieraus ergibt sich folgender Dreisatz: Eingangsspannung = $\frac{\textit{Dezimalwert} \times 39V}{1024}$
	Beispiel: Eingabe: <STX>140503<ETX>, Abfrage Modul (Knoten) 20 Antwort: <STX>84 02<ETX> = 02 84 = Dezimalwert = 644 Eingangsspannung = $\frac{644 \times 39V}{1024}$ = 24,53V	
Ausgangsspg.	<STX>nn 0502<ETX> Modul (Knoten) — Kommando	<STX> XX YY <ETX> YY = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB, LSB (Hex) = Umwandlung in *Dezimalwert* Der maximale Messwert ist 1024 und entspricht 39V. Hieraus ergibt sich folgender Dreisatz: Ausgangsspannung = $\frac{\textit{Dezimalwert} \times 39V}{1024}$
	Beispiel: Eingabe: <STX>210502<ETX>, Abfrage Modul (Knoten) 33 Antwort: <STX>87 02<ETX> = 02 87 = Dezimalwert = 647 Ausgangsspannung = $\frac{647 \times 39V}{1024}$ = 24,64 V	
Strommessung	<STX>nn 0511<ETX> Modul (Knoten) — Kommando	<STX> XX YY <ETX> YY = MSB (höchstwertige Bit) XX = LSB (niederwertigste Bit) MSB, LSB (Hex) = Umwandlung in *Dezimalwert* Der maximale Messwert ist 1024 und entspricht 32,75A. Hieraus ergibt sich folgender Dreisatz: Strom = $\frac{\textit{Dezimalwert} \times 32{,}75A}{1024}$
	Beispiel: Eingabe: <STX>0F0511<ETX>, Abfrage Modul (Knoten) 15 Antwort: <STX>1F 00<ETX> = 00 1F = Dezimalwert = 31 Strom = $\frac{31 \times 32{,}75A}{1024}$ = 0,99 A	

<table>
<tr><th>Abfrage</th><th>Eingabe - Kommando</th><th>Antwort</th></tr>
<tr><th></th><th colspan="2"><STX> = Strg+B / <ETX> = Strg+C</th></tr>
<tr><td rowspan="2">Kennlinien-einstellungen</td><td><STX>nn 051D<ETX>
Modul (Knoten) – nn
Kommando – 051D</td><td><STX> XX Y Y <ETX>
(binär) 0 0 0 0 0 0 0 0
00: m = 2
01: m = 3
10: m = 5
11: m = 9
0: n = 3-Cubic
1: n = 4-Quad
000: Tp(s) = 1,25
001: Tp(s) = 2,5
010: Tp(s) = 5,0
011: Tp(s) = 10,0
100: Tp(s) = 20,0
101: Tp(s) = 40,0
110: Tp(s) = 80,0
reserviert
XX: Umwandlung in Dezimalwert,
Der maximale Messwert ist 256 und entspricht 32,75A. Hieraus ergibt sich folgender Dreisatz:
$Iq = \frac{(256 - Dezimalwert) \times 32{,}75A}{256}$</td></tr>
<tr><td colspan="2"><u>Beispiel:</u> Eingabe: <STX>0A051D<ETX>, Abfrage Modul (Knoten) 10
Antwort: <STX>99 1F<ETX>
1F: 00 01 1 1 11
11: m = 9
1: n = 4-Quad
011: Tp(s) = 10,0
99: = Dezimalwert = 153
$Iq = \frac{(256 - 153) \times 32{,}75A}{256} = 13{,}18\ A$</td></tr>
</table>

2.4.2 Zustand AN / AUS schreiben

<table>
<tr><th>Abfrage</th><th>Eingabe - Kommando</th><th>Antwort</th></tr>
<tr><th></th><th colspan="2"><STX> = Strg+B / <ETX> = Strg+C</th></tr>
<tr><td rowspan="2">Zustand AN / AUS</td><td><STX> nn 84 xx <ETX>
Modul (Knoten) – nn
Kommando (schreiben) – 84
Befehl (AN / AUS) – xx
XX: 00 = Modul ausschalten
01 = Modul einschalten</td><td>Als Antwort wird der neue Betriebszustand zurück gegeben. Die Antwort ist gemäß “Modulzustand/Konfiguration“ Seite 25 zu lesen.</td></tr>
<tr><td colspan="2"><u>Beispiel:</u>

Eingabe: <STX>088401<ETX>, Einschalten Modul (Knoten) 8
Antwort: <STX>09 25<ETX>, 09 = Modul ist eingeschaltet
25 = Strombereich 6 (5+1), Charakteristik 3 (2+1)</td></tr>
</table>

2.5 Kommunikation über CANopen

Auf dem CAN-Bus verhält sich das Gateway wie ein CANopen-Knoten gemäß DS301 mit einem Profil in Anlehnung an die DS401 (Generic I/O).

2.5.1 Begriffe

CANopen	CANopen ist ein auf CAN basierendes Protokoll, welches ursprünglich für industrielle Steuerungssysteme entwickelt wurde. www.can-cia.org
COB	Communication Object
Datenrate	Die Datenrate ist die Anzahl der Daten die in einer bestimmten Zeit übertragen werden können.
EDS-Datei	Electronic Data Sheet - elektronisches Datenblatt Wird vom Hersteller eines CANopen-Gerätes bereitgestellt. Es hat ein standardisiertes Format für die Beschreibung von Geräten. Die EDS-Datei beinhaltet Informationen über die Beschreibung der Datei, allgemeine Geräteinformationen und die Beschreibung der unterstützten Objekte.
Emergency-Id	Emergency Data Object - Notfalldaten
ID	Identifier
Knotennummer	Innerhalb eines CANopen-Netzwerkes wird jedes Gerät über seine Knotennummer (Node-ID) identifiziert. Die erlaubten Knotennummern liegen im Bereich von 1-127 und dürfen nur einmal innerhalb eines Netzwerkes vorkommen.
LEN	Length - Länge der Daten
NMT	Network Management (Master) - Netzwerkmanagement
PDO	Process Data Objects - Prozess-Daten-Objekte Die PDOs dienen zur Übertragung der Prozessdaten. Im ‘Receive’-PDO (RxPDO) werden Prozessdaten vom LOCC-Box Gateway empfangen. Im ‘Transmit’-PDO (TxPDO) sendet das Gateway Daten auf das CANopen Netz.
PDO-Mapping	Die Größe eines PDOs kann bis zu 8 Byte betragen. Es kann benutzt werden, um mehrere Anwendungsobjekte zu transportieren. Das PDO-Mapping beschreibt die Festlegung über die Anordnung der Anwendungsobjekte innerhalb des Datenfeldes des PDOs.
RTR	Remote Transmit Request
Rx	Receive - empfangen
SDO	Service Data Object - Service-Daten-Objekte (Parameter) Die SDOs dienen zur Übertragung von modulinternen Konfigurations- und Parameterdaten wie z.B. der Ausgangsstrom. Im Gegensatz zu den PDOs werden die SDO-Nachrichten bestätigt. Einer Schreib- oder Leseanforderung auf ein Daten-Objekt folgt immer ein Bestätigungs- oder Fehler-Telegramm.
Sync	Sync (frame) Telegramm - Synchronisations-Identifier
Tx	Transmit - senden



2.5.2 Beschreibungsdatei

Die zu verwendende EDS-Datei (Electronic Data Sheet) befindet sich in der Datei „LOCC-Pads_x.x.x.x.zip“, welche auf der Lütze Homepage als kostenloser Download zur Verfügung steht.

2.5.3 NMT-Boot-up

Das Gateway kann mit dem im CiA-Draft Standard 301 in Kapitel 9.4 beschriebenen ‘Minimal - Boot-up’ initialisiert werden. Als Beispiel wird das Gateway mit der Node-ID 5 verwendet.

Befehl	Identifer [Hex]	Befehlscode	Node-ID vom Gateway
NMT - “Operation Mode“	0000	01	05$_h$
NMT - “Pre-Operation Mode“	0000	80	05$_h$

Im Operation Mode werden zusätzlich Prozess-Daten-Objekte (PDO) unterstützt!

2.5.4 Das CANopen-Objektverzeichnis

Das Objektverzeichnis ist im wesentlichen eine (sortierte) Gruppierung von Objekten, auf die über das Netzwerk zugegriffen werden kann. Jedes Objekt in diesem Verzeichnis wird mit einem 16-Bit-Index adressiert. In den Objektverzeichnissen wird der Index in hexadezimaler Form angegeben.

Der Index kann 16-Bit-Parameter nach der CANopen-Spezifikation (CiA DS-301) oder ein herstellerspezifischer Code sein. Anhand der höherwertigen Bits des Index wird festgelegt, zu welcher Objektklasse der Parameter gehört. Zum Objektverzeichnis gehören unter anderem:

Index [Hex]	Objekt	Beispiel
0001 … 009F	Definition von Datentypen	-
1000 … 1FFF	Communication Profile Area	1001$_h$: Fehler-Register
2000 … 5FFF	Manufacturer Specific Profile Area	2000$_h$: Modul-Typ
6000 … 9FFF	Standardised Device Profile Area	nach Anwendungsprofil DS-40x
A000 … FFFF	reserviert	-

2.5.4.1 Zugriff auf das Objektverzeichnis über SDOs

Die SDOs (Service Data Objects) dienen zum Zugriff auf das Objektverzeichnis eines Gerätes. Ein SDO stellt daher einen ‘Kanal’ zum Zugriff auf die Parameter des Gerätes dar. Der Zugriff über diesen Kanal ist beim Gateway im Zustand operational und pre-operational möglich.
Die SDOs werden auf der ID ‘**600$_h$ + Node-ID**’ übertragen (Request). Der Empfänger quittiert die Parameter auf der ID ‘**580$_h$ + Node-ID**’ (Response).

Ein SDO ist wie folgt aufgebaut:

Identifier (600$_h$+ Node-ID)	Befehls-code	Index		Sub-Index	LSB	Datenfeld		MSB
		(low)	(high)					

Beispiel - Senden (Request):

605$_h$ (Node-ID 5 Gateway)	40$_h$ (read)	00$_h$	20$_h$	01 (LOCC-Box Knoten 1)	00$_h$	00$_h$	00$_h$	00$_h$
		(Index=2000$_h$) (Transmit-SDO-Modultyp)						

Beispiel - Empfangen (Response):

585$_h$ (Node-ID 5 Gateway)	4F$_h$ (1 Byte)	00$_h$	20$_h$	01 (LOCC-Box Knoten 1)	01$_h$	00$_h$	00$_h$	00$_h$
		(Index=2000$_h$) (Receive-SDO-Modultyp)			Wert = 00 00 00 01 (MSB … LSB)			

Beschreibung der SDOs:

Identifier: Die Parameter werden auf der ID ‘600_h + Node-ID’ übertragen (Request). Der Empfänger quittiert die Parameter auf der ID ‘580_h + Node-ID’ (Response).

Befehlscode: Der gesendete Befehlscode setzt sich unter anderem aus dem Command Specifier und der Länge zusammen. Häufig benötigte Kombinationen sind z.B.:

$40_h = 64_d$: Read Request, d.h. ein Parameter soll gelesen werden
$23_h = 35_d$: Write Request mit 32 Bit Daten, d.h. ein Parameter soll gesetzt werden

Das Gateway antwortet auf jedes empfangene Telegramm mit einem Antworttelegramm. Dies kann folgende Befehlscodes enthalten:

$43_h = 67_d$: Read Response mit 32 Bit Daten, dieses Telegramm enthält den gewünschten Parameter
$60_h = 96_d$: Write Response, d.h. ein Parameter wurde erfolgreich gesetzt
$80_h = 128_d$: Error Response, d.h. das CAN-Modul meldet einen Fehler.

Die folgende Tabelle gibt eine Übersicht über häufig verwendete Befehlscodes. Die Kommando-Frames müssen immer 8 Daten-Bytes beinhalten. Hinweise zur Syntax und weitere Befehlscodes sind in der CiA DS-301 enthalten.

Kommando	für Anzahl Datenbytes	Befehlscode [Hex]
Write Request (Initiate Domain Download)	1 2 3 4	2F 2B 27 23
Write Response (Initiate Domain Download)	-	60
Read Request (Initiate Domain Upload)	-	40
Read Response (Initiate Domain Upload)	1 2 3 4	4F 4B 47 43
Error Response (Abort Domain Transfer)	-	80

Index, Sub-Index Der Index und der Sub-Index werden in den Kapiteln “Device Profile Area, 4.4.7” und “Manufacturer Specific Profile Area, 4.4.8” dieses Handbuches beschrieben.

Datenfeld … Das maximal 4 Byte lange Datenfeld ist grundsätzlich nach der Regel niederwertiges Byte (LSB) zuerst, höherwertiges Byte (MSB) zuletzt aufgebaut. Dabei steht das niederwertige Byte immer in ‘Data 1’, bei 16-Bit-Werten steht das höchstwertige Byte (Bits 8...15) in ‘Data 2’, und bei 32-Bit-Werten steht das MSB (Bits 24...31) in ‘Data 4’.

2.5.4.2 Fehlercodes des SDO-Domain-Transfers

Die folgenden Fehlercodes können zur Anwendung kommen (gemäß CiA DS-301, Kapitel "Abort SDO Transfer Protocol"):

Fehlercode [Hex]	Erläuterung
05030000	Toggle Bit unverändert
05040001	falscher Command Specifier
06010002	Schreibzugriff ist hier falsch
06020000	falscher Index
06040041	Objekt kann nicht auf PDO gemapped werden
06040043	Parameter inkompatibel
06060000	kein Zugriff wegen Hardware Fehler
06070010	falsche Anzahl Datenbytes
06070012	Länge des Service-Parameters ist zu groß
06070013	Länge des Service-Parameters ist zu klein
06090011	falscher Sub-Index
06090030	gesendeter Parameter außerhalb des zul. Wertebereiches
08000000	nicht definierte Fehlerursache
08000020	Daten können nicht übertragen o. gespeichert werden
08000022	wegen des aktuellen Device States können Daten nicht übertragen oder gespeichert werden
08000024	Zugriff auf Flash fehlgeschlagen

2.5.5 Übersicht der verwendeten CANopen-Identifier

Funktion	Identifier [Hex]	Bemerkungen
Netzwerkmanagement	0	NMT (Operation-/ Pre-Operation Mode)
SYNC	80	Sync an alle, (konfigurierbar über Objekt 1005_h)
Emergency Message	80 + *Node-ID*	konfigurierbar über Objekt 1014_h
Tx-PDO1	180 + *Node-ID*	PDO1 vom Gateway (Tx) (Objekt 1800_h)
Tx-PDO2	280 + *Node-ID*	PDO2 vom Gateway (Tx) (Objekt 1801_h)
Tx-PDO3	380 + *Node-ID*	PDO3 vom Gateway (Tx) (Objekt 1802_h)
Tx-PDO4	480 + *Node-ID*	PDO4 vom Gateway (Tx) (Objekt 1803_h)
Rx-PDO1	200 + *Node-ID*	PDO1 zum Gateway (Rx) (Objekt 1400_h)
Rx-PDO2	300 + *Node-ID*	PDO2 zum Gateway (Rx) (Objekt 1401_h)
Rx-PDO3	400 + *Node-ID*	PDO3 zum Gateway (Rx) (Objekt 1402_h)
Rx-PDO4	500 + *Node-ID*	PDO4 zum Gateway (Rx) (Objekt 1403_h)
Tx-SDO	580 + *Node-ID*	SDO vom Gateway (Tx)
Rx-SDO	600 + *Node-ID*	SDO zum Gateway (Rx)
Node Guarding	700 + *Node-ID*	konfigurierbar über Objekt $100E_h$

Node-ID: Eingestellte CANopen-Adresse [1_h...$7F_h$]



2.5.5.1 Einstellung der COB-ID

Die COB-IDs die einstellbar sind (außer der von SYNC), werden zunächst von der Einstellung der Node-ID über die Software LOCC-Pads (Kapitel 3.4.2) abgeleitet. Wurden die COB-IDs über SDO beschrieben, gilt diese Einstellung, auch wenn in LOCC-Pads eine andere Node-ID eingestellt wird. Um die Default-COB-ID wieder von LOCC-Pads zu übernehmen, müssen die Comm Defaults oder alle Defaults geladen werden (Objekt 1011h)

2.5.6 Setzen und Lesen der LOCC-Box-Net

2.5.6.1 Meldung der LOCC-Box-Net

Die Übertragungsarten können unterschieden werden in:

- *azyklisch, synchron:* Die Sendung erfolgt nach dem Empfang einer SYNC-Message (PDO -Übertragungstyp 0), wenn sich die Daten geändert haben
- *zyklisch, synchron:* Die Sendung erfolgt, nachdem jeweils eine bestimmte Anzahl SYNC-Messages empfangen wurden (PDO-Übertragungstyp 1...240).
- synchron, remote request: Der Zustand der Eingänge wird mit jeder SYNC-Message gespeichert und nach dem Empfang eines RTR-Frames gesendet (PDO-Übertragungstyp 252).
- asynchron, remote request: Nach dem Empfang eines RTR-Frames wird der letzte ermittelte Zustand der Eingänge gesendet (PDO-Übertragungstyp 253).
- ereignisgesteuert, asynchron: Die Sendung erfolgt, wenn sich der Zustand bestimmter LOCC-Box-Net Module ändert (PDO-Übertragungstyp 254, 255).

2.5.6.2 Einschalten der LOCC-Box-Net

Die LOCC-Box-Net Module werden gesetzt, sobald ein Objekt zum Einschalten empfangen wurde (z.B. Objekt 6200h oder Rx-PDO).

2.5.6.3 Unterstützte Übertragungsarten nach DS-301

Übertragungs-typ	**PDO - Übertragung**					**Unterstützt vom Gateway**
	zyklisch	azyklisch	synchron	asynchron	RTR	
0		X	X			X
1 … 240	X		X			X
241 … 251	reserviert					-
252			X		X	X
253				X	X	X
254				X	X	X
255				X	X	X

2.5.7 Implementierte PDO's

Das Gateway unterstütz Tx- und Rx-PDO's im aktivierten "Operation Mode". Siehe Kapitel 4.4.2.

2.5.7.1 Tx – PDO

Das Transmit PDO sendet vom Gateway die Information, welches Modul an- oder ausgeschaltet ist. Es werden folgende Identifier verwendet:

- 180 + *Node-ID,* für die Knotennummern 1 – 64
- 280 + *Node-ID,* für die Knotennummer 65 – 128
- 380 + *Node-ID,* für die Knotennummer 129 – 192
- 480 + *Node-ID,* für die Knotennummer 193 – 254

Mit jeder Zustandsänderung wird das entsprechende PDO gesendet. Zustandsänderungen sind:

- Module wurden ein- oder ausgeschaltet
- Module haben ausgelöst
- Module wurden quittiert
- Der Ausgangsstrom überschreitet 90% vom eingestellten Strombereich
- Der Ausgangsstrom unterschreitet 90% vom eingestellten Strombereich

Beispiel:
An einem Modul mit der Knotennummer 1 – 64 wurde eine Zustandsänderung erkannt. Das Gateway mit der Knotennummer 5 sendet ein PDO auf dem Identifier 185_h (180 + Node-ID) an die SPS.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			LSB	…	…	…	…	…	…	MSB
185_h	0_h	8_h	FF_h	00_h	$F9_h$	$0A_h$	$9B_h$	01_h	30_h	00_h
			gelesener Wert: 00 30 01 9B 0A F9 00 FF_h							

Es werden 8 Datenbytes übertragen und binär ausgewertet. 0 = Zustand AUS, 1 = Zustand AN

Binär: 0000 0000 0011 0000 0000 0001 1001 1011 0000 1010 1111 1001 0000 0000 1111 1111

Modul: 64 … 57 56 … 49 48 … 41 40 … 33 32 … 25 24 … 17 16 … 9 8 … 1

2.5.7.2 Rx – PDO

Das Receive PDO erlaubt dem Gateway das Empfangen von Informationen, welches Modul bzw. Knoten an- oder ausgeschaltet werden soll. Es werden folgende Identifier verwendet:

- 200 + *Node-ID,* für die Knotennummer 1 – 64
- 300 + *Node-ID,* für die Knotennummer 65 – 128
- 400 + *Node-ID,* für die Knotennummer 129 – 192
- 500 + *Node-ID,* für die Knotennummer 193 – 254

Beispiel:
Das Gateway-Modul Nr. 5 empfängt die Informationen für die Knotennummern 1 – 64. Es werden 8 Datenbytes übertragen. 0 = Zustand AUS, 1 = Zustand AN

Binär: 0000 0000 0011 0000 0000 0001 1001 1011 0000 1010 1111 1001 0000 0000 1111 1111

Modul: 64 … 57 56 … 49 48 … 41 40 … 33 32 … 25 24 … 17 16 … 9 8 … 1

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			LSB	…	…	…	…	…	…	MSB
205_h	0_h	8_h	FF_h	00_h	$F9_h$	$0A_h$	$9B_h$	01_h	30_h	00_h
			geschriebener Wert: 00 30 01 9B 0A F9 00 FF_h							

HINWEIS

Ausgelöste Module werden durch das Ausschalten nicht quittiert. Der Befehl Einschalten überspringt in diesem Fall das Quittieren und schaltet das Modul sofort zu.

2.5.8 Implementierte CANopen – Objekte (1000_h … $1FFF_h$)

Eine detaillierte Beschreibung der Objekte ist in der CiA DS-301 nachzulesen.

Index	Sub-Index	Beschreibung	Datentyp	R/W	Default-Wert	Beispiel im Kapital
1000_h	-	Device type	unsigned 32	ro	00030191_h	4.4.7.1
1001_h	-	Error Register	unsigned 8	ro	00_h	4.4.7.2
1003_h	16	Error-Field	unsigned 32	ro	00_h	4.4.7.3
1005_h	-	COB-ID-Sync	unsigned 32	rw	80_h	4.4.7.4
1006_h	-	Communication cycle periode	unsigned 32	ro	0000000_h	4.4.7.5
1008_h	-	Device Name	visible string	ro	LOCC-Box-Net	4.4.7.6
1009_h	-	Hardware Version	visible string	ro	x.y	4.4.7.7
$100A_h$	-	Software Version	visible string	ro	x.y	4.4.7.8
$100C_h$	-	Guard time	unsigned 16	rw	0000_h	4.4.7.9
$100D_h$	-	Life time factor	unsigned 8	rw	00_h	4.4.7.9
$100E_h$	-	Nodeguard COB-ID	unsigned 32	rw	700_h + Node-ID	4.4.7.10

1010_h	1	Store Parameters	unsigned 32	rw	1_h	4.4.7.11
1011_h	1	Restore Default Parameters	unsigned 32	rw	1_h	4.4.7.12
1014_h	-	COB-ID emergency message	unsigned 32	rw	80_h + Node-ID	4.4.7.13
1015_h	-	Inhibit time emergency	unsigned 16	rw	00_h	4.4.7.14
1016_h	1	Consumer heartbeat time	unsigned 32	rw	00_h	4.4.7.15
1017_h	-	Producer heartbeat time	unsigned 16	rw	00_h	4.4.7.16
1018_h	4	Identity object	unsigned 32	ro	-	4.4.7.18
	1	Vendor –ID			64_h	
	2	Product code, dezimal			716459	
	3	Revision number			x.y	
	4	Serial number			0_h	
1020_h	2	Verify configuration	unsigned 32	rw	00_h	4.4.7.18
1029_h	1	Error behaviour	unsigned 8	rw	00_h	4.4.7.19

Index	Sub-Index	Beschreibung	Datentyp	R/W	Beispiel im Kapitel
1400_h	2	1. Receive PDO-Parameter	PDO CommPar	rw	4.4.7.20
1401_h	2	2. Receive PDO-Parameter	PDO CommPar	rw	4.4.7.20
1402_h	2	3. Receive PDO-Parameter	PDO CommPar	rw	4.4.7.20
1403_h	2	4. Receive PDO-Parameter	PDO CommPar	rw	4.4.7.20
1600_h	8	1. Receive PDO-Mapping	PDO Mapping	ro	4.4.7.21
1601_h	8	2. Receive PDO-Mapping	PDO Mapping	ro	4.4.7.21
1602_h	8	3. Receive PDO-Mapping	PDO Mapping	ro	4.4.7.21
1603_h	8	4. Receive PDO-Mapping	PDO Mapping	ro	4.4.7.21
1800_h	5	1. Transmit PDO-Parameter	PDO CommPar	rw	4.4.7.22
1801_h	5	2. Transmit PDO-Parameter	PDO CommPar	rw	4.4.7.22
1802_h	5	3. Transmit PDO-Parameter	PDO CommPar	rw	4.4.7.22
1803_h	5	4. Transmit PDO-Parameter	PDO CommPar	rw	4.4.7.22
$1A00_h$	8	1. Transmit PDO-Mapping	PDO Mapping	ro	4.4.7.23
$1A01_h$	8	2. Transmit PDO-Mapping	PDO Mapping	ro	4.4.7.23
$1A02_h$	8	3. Transmit PDO-Mapping	PDO Mapping	ro	4.4.7.23
$1A03_h$	8	4. Transmit PDO-Mapping	PDO Mapping	ro	4.4.7.23

ro = nur lesen
rw = lesen und schreiben

2.5.8.1 Device Typ (1000_h)

Index	Name	Datentyp	R/W	Default-Wert
1000_h	*Device typ*	Unsigned 32	ro	0003 0191_h

Der Wert des *Device typ* beträgt: 0003.0191_h

Digitaler Ein-/Ausgang: 0003_h
Digitale Profilnummer: 0191_h

Auslesen des Device Typ

Der CANopen Master sendet unter dem Identifier '605_h' (600_h + Node-ID) den Read Request an das Gateway mit der Modul-Nr. 5 (Node-ID=5_h):

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	40_h	00_h	10_h	00_h	00_h	00_h	00_h	00_h
			Read Request	Index=1000_h						

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '585_h' (580_h + Node-ID) mit dem Wert des Device Type:

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
585_h	0_h	8_h	43_h	00_h	10_h	00_h	91_h	01_h	03_h	00_h
			Read Response	Index=1000_h			gelesener Wert: 00 03 01 91_h			

Der gelesene Wert entspricht dem Device Typ: 0003.0191_h (siehe oben)

2.5.8.2 Error Register (1001_h)

Index	Name	Datentyp	R/W	Default-Wert
1001_h	*Error Register*	Unsigned 8	ro	00_h

Das Gateway nutzt das Error-Register, um Fehlermeldungen anzuzeigen. Folgende Bits des Error-Registers werden zurzeit unterstützt:

Bit	Bedeutung
0	*Generic*
1	-
2	-
3	-
4	*communication error (overrun, error state)*
5	-
6	-
7	-

Die nicht unterstützten Bits (-) werden immer als ‘0’ zurückgegeben. Liegt ein Fehler vor, so ist das entsprechende Fehler-Bit auf ‘1’ gesetzt.

Folgende Fehlermeldung wird unterstützt:

11_h communication error



2.5.8.3 Pre-defined Error Field (1003$_h$)

Index	Name	Datentyp	R/W	Default-Wert
1003$_h$	*Pre-defined error field*	unsigned 32	ro	00$_h$

Im *pre-defined error field* wird eine Liste der zuletzt aufgetretenen Fehler gespeichert. Der Sub-Index 0 enthält die aktuelle Anzahl der in der Liste gespeicherten Fehler. Unter Sub-Index 1 wird der zuletzt aufgetretene Fehler abgelegt. Tritt ein neuer Fehler auf, so wird der vorhergehende Fehler auf Sub-Index 2 gespeichert und der neue Fehler unter Sub-Index 1 usw.. So entsteht eine Liste mit der Fehler-Historie.
Der Fehlerspeicher ist wie ein Ringspeicher aufgebaut. Ist er gefüllt, wird der älteste Eintrag gelöscht, um Platz für den aktuellen Eintrag zu schaffen.
Dieses Modul unterstützt maximal 10 Fehlereinträge. Beim Eintreten des 11. Fehlers wird der älteste Fehlereintrag gelöscht. Zum Löschen der gesamten Fehler-Liste ist der Sub-Index ‘0’ auf ‘0’ zu setzen. Dies ist der einzig zulässige Schreibzugriff auf das Objekt.
Mit jedem neuen Eintrag in die Liste sendet das Modul ein Emergency-Frame, um den Fehler mitzuteilen.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert [Dez]	Datentyp	R/W
1003$_h$	0	*no_of_errors_in_list*	0, 1…10	-	unsigned 8	rw
	1	*error-code n*	0…FFFFFFFF	-	unsigned 32	ro
	2	*error-code n-1*	0…FFFFFFFF	-	unsigned 32	ro
	:	:	:	:	:	ro
	16	*error-code n-16*	0…FFFFFFFF	-	unsigned 32	ro

Bedeutung der Variablen

no_of_errors_in_list - enthält die aktuelle Anzahl der in der Liste eingetragenen Fehler-Codes, n = Nummer des zuletzt aufgetretenen Fehlers
- zum Löschen der Fehlerliste ist diese Variable auf ‘0’ zu setzen
- ist no_of_errors_in_list …0, so wird das Error-Register (Object 1001$_h$) gesetzt

error-code x Der 32-Bit lange Fehler-Code setzt sich aus dem im DS-301, Tabelle 21 aufgeführten CANopen-Emergency-Error-Code und den von Lütze definierten Fehler-Codes (Manufacturer-Specific Error Field) zusammen.

Bit	31 … … 16	15 … … 0
Inhalt	*manufacturer-specific error field*	*emergency-error-code*

manufacturer-specific error field: immer ‘00’

emergency-error-code: Es werden die folgenden Fehler-Codes unterstützt:

8001$_h$ – No SYNC received
8120$_h$ - CAN in Error Passive Mode
8130$_h$ - Lifeguard Error
8140$_h$ - Recovered from “Bus Off”

Emergency Message Die Daten des vom Gateway gesendeten Emergency-Frames sind wie folgt aufgebaut:

Byte	0	1	2	3	4	5	6	7
Inhalt	*emergency-error-code*		*error-register* 1001$_h$	*no_of_errors_in_list* 1003,00$_h$				

2.5.8.4 COB-ID of SYNC-Message (1005_h)

Index	Name	Datentyp	R/W	Default-Wert
1005_h	*COB-ID of SYNC-Message*	unsigned 32	rw	80_h

Struktur des Parameters:

Bit-No.	Wert	Bedeutung
31 (MSB)	-	do not care
30	0/1	0: keine SYNC Erzeugung, 1: Modul erzeugt SYNC
29	0	immer 0, da 11-Bit-ID
28…11	0	immer 0, da keine 29-Bit-ID unterstützt werden
10…0 (LSB)	x	Bit 0...10 der SYNC-COB-ID

Der Identifier kann Werte zwischen 0...$7FF_h$ annehmen.

2.5.8.5 Communication Cycle Period (1006_h)

Index	Name	Datentyp	R/W	Default-Wert
1006_h	*Communication Cycle Period*	unsigned 32	rw	80_h

Die Communication Cycle Periode bestimmt das Zeitintervall, in dem SYNC-Telegramme gesendet werden (wenn in Objekt 1005_h SYNC-Erzeugung eingeschaltet wurde), bzw. SYNC-Telegramme erwartet werden.

2.5.8.6 Device Name (1008_h)

Index	Name	Datentyp	R/W	Default-Wert
1008_h	*Device Name*	visible string	ro	LOCC-Box-Net

Eine ausführliche Beschreibung des Domain Uploads ist der CiA DS-202-2 (CMS-Protocol Specification) zu entnehmen.

2.5.8.7 Hardware Version (1009_h)

Index	Name	Datentyp	R/W	Default-Wert
1009_h	*Hardware Version*	visible string	ro	x.y

Das Lesen der Hardware-Version erfolgt ähnlich wie das Lesen des Manufacturer Device Names über das Domain Upload Protokoll. Eine ausführliche Beschreibung des Uploads ist der CiA DS-202-2 (CMS-Protocol Specification) zu entnehmen.

2.5.8.8 Software Version ($100A_h$)

Index	Name	Datentyp	R/W	Default-Wert
$100A_h$	*Software Version*	visible string	ro	x.y

Das Lesen der Software-Version erfolgt ähnlich wie das Lesen des Manufacturer Device Names über das Domain Upload Protokoll. Eine ausführliche Beschreibung des Uploads ist der CiA DS-202-2 (CMS-Protocol Specification) zu entnehmen.

2.5.8.9 Guard time (100C$_h$) und Life time factor (100D$_h$)

Die LOCC-Box unterstützt das Node-Guarding oder alternativ die Heartbeat-Funktion (siehe Seite 41). Guard Time und Life Time Factor werden zusammen ausgewertet. Die Multiplikation beider Werte ergibt die Life Time. Die Guard Time wird in Millisekunden angegeben.

Index	Name	Datentyp	R/W	Default-Wert
100C$_h$	*Guard time*	unsigned 16	rw	0000$_h$
100D$_h$	*Life time factor*	unsigned 8	rw	00$_h$

2.5.8.10 Node Guard COB-ID (100E$_h$)

Das Modul unterstützt nur 11-Bit-Identifier.

Index	Name	Datentyp	R/W	Default-Wert
100E$_h$	*Node guard COB-ID*	unsigned 32	rw	Node-ID + 700$_h$

Struktur des Parameters *Node guarding identifier:*

Bit-No.	Bedeutung
31 (MSB) 30	reserviert
29…11	immer 0, da keine 29-Bit-ID unterstützt werden
10…0 (LSB)	Bit 0...10 des Node Guarding Identifiers

Der Identifier kann Werte zwischen 1...7FF$_h$ annehmen.

2.5.8.11 Store Parameters (1010$_h$)

Index	Name	Datentyp	R/W	Default-Wert
1010$_h$	*Store Parameters*	unsigned 32	rw	1$_h$

Mit diesem Kommando werden die Parameter im EEPROM gespeichert.

Beim Schreiben des Index muss die unten angegebene Byte-Folge gesendet werden. Das Lesen des Index gibt Informationen über die implementierten Speicher-Funktionen zurück (näheres hierzu siehe CiA DS-301).

Index	Sub-Index	Beschreibung	Wertebereich [Hex]	Datentyp	R/W
1010$_h$	0	*number_ of_entries*	4$_h$	unsigned 8	ro
	1	*save_all_parameters* (Objekte 1000$_h$…9FFF$_h$)	no default, write: 65 76 61 73$_h$ (=ASCII: ‘e‘ ‘v‘ ‘a‘ ‘s‘)	unsigned 32	rw

Parameter die abgespeichert bzw. geladen werden können:

- Communication Parameter der Objekte 1005$_h$ ... 1029$_h$
- Application Parameter der Objekte 6000$_h$ ... 6100$_h$
- Manufacturer Specific Parameter der Objekte 2220$_h$ ... 2310$_h$



2.5.8.12 Restore Default Parameters (1011_h)

Index	Name	Datentyp	R/W	Default-Wert
1011_h	*Restore default parameters*	unsigned 32	rw	1_h

Mit diesem Kommando werden die bei der Auslieferung aktiven Default-Parameter wieder aktiviert. Alle individuellen Einstellungen, die im EEPROM gespeichert worden sind, gehen verloren. Es wird nur das Kommando 'Restore all Parameters' unterstützt. Beim Schreiben des Index muss die unten angegebene Byte-Folge gesendet werden. Das Lesen des Index gibt Informationen über die implementierten Restore-Funktionen zurück (näheres hierzu siehe CiA DS-301).

Index	Sub-Index	Beschreibung	Wertebereich [Hex]	Datentyp	R/W
1011_h	0	*number_ of_entries*	4_h	unsigned 8	ro
	1	*load_all_default_parameters* (Objekte 1000_h…$9FFF_h$)	no default, write: 64 61 6F $6C_h$ (=ASCII: 'd' 'a' 'o' 'l')	unsigned 32	rw

Parameter die abgespeichert bzw. geladen werden können:

- Communication Parameter der Objekte 1005_h ... 1029_h
- Application Parameter der Objekte 6000_h ... 6100_h
- Manufacturer Specific Parameter der Objekte 2220_h ... 2310_h

2.5.8.13 COB_ID Emergency Message (1014_h)

Index	Name	Datentyp	R/W	Default-Wert
1014_h	*COB_ID emergency message*	unsigned 32	rw	80_h + Node-ID

Dieses Objekt bestimmt die COB-ID der Emergency Message (EMCY).

Die Struktur des Objektes ist in der folgenden Tabelle beschrieben:

Bit-No.	Wert	Bedeutung
31 (MSB)	0/1	0: EMCY existiert / ist gültig, 1: kein EMCY / EMCY ist nicht gültig
30	0	reserviert (immer 0)
29	0	immer 0, da 11-Bit ID
28…11	0	immer 0, da keine 29-Bit-ID unterstützt werden
10…0 (LSB)	x	Bit 0...10 des COB-ID

Der Identifier kann Werte zwischen 0...$7FF_h$ annehmen.

2.5.8.14 Inhibit Time Emergency (1015_h)

Index	Name	Datentyp	R/W	Default-Wert
1015_h	*Inhibit time emergency*	unsigned 16	rw	00_h

Mit diesem Objekt kann die Inhibit Time für die Emergency-Nachricht festgelegt werden. Die Zeit wird als ein Vielfaches von 100µs angegeben.

2.5.8.15 Consumer Heartbeat Time (1016_h)

Index	Name	Datentyp	R/W	Default-Wert
1016_h	*Consumer Heartbeat Time*	unsigned 32	rw	00_h

Zur gegenseitigen Überwachung der CANopen-Module (insbesondere zum Erkennen von Verbindungsausfällen) kann die Heartbeat-Funktion genutzt werden. Im Gegensatz zum Node Guarding/Life Guarding kommt die Heartbeat-Funktion ohne RTR-Frames aus.
Ein Modul, der sogenannte Heartbeat-Producer sendet auf dem Node-Guarding-Identifier (siehe Object$100E_h$) zyklisch eine Heartbeat-Nachricht auf dem CAN-Bus. Ein oder mehrere Heartbeat-Consumer empfangen die Nachricht. Die Nachricht muss innerhalb der auf dem Heartbeat-Consumergespeicherten Heartbeat-Time empfangen werden, sonst wird auf dem Heartbeat-Consumer-Modul ein Heartbeat-Event ausgelöst. Auf dem Gateway löst der Heartbeat-Event einen Heartbeat-Error aus. Jedes Modul kann Heartbeat-Producer und Heartbeat-Consumer sein. Das Gateway unterstützt in einem CAN-Netz maximal einen Heartbeat-Producer.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert [Dez]	Datentyp	R/W
1016_h	0	*number_ of_entries*	1	1	unsigned 8	ro
	1	*Consumer-heartbeat_time*	0 … 007FFFFF	0	unsigned 32	rw

Bedeutung der Variablen:

consumer-heartbeat_time_x

	consumer-heartbeat_time_x		
Bit	31 … … 24	23 … … 16	15 … … 0
Belegung	reserved (immer “0”)	Node-ID (unsigned 8)	Heartbeat_time (unsigned 16)

Node-ID Node-ID des zu überwachenden Heartbeat-Producers.

heartbeat_time Zeit, innerhalb der sich der zu überwachende Heartbeat-Producer auf dem Node-Guarding-ID melden muss, damit kein Heartbeat-Event ausgelöst wird. Diese Consumer-Heartbeat-Time muss immer größer sein als die Producer-Heartbeat-Time.

2.5.8.16 Producer Heartbeat Time (1017_h)

Index	Name	Datentyp	R/W	Default-Wert
1017_h	*Producer heartbeat time*	unsigned 16	rw	00_h

Hier wird die Zeit eingetragen, mit der das Gateway zyklisch einen Heartbeat-Frame auf dem Node-Guarding-ID sendet. Wird für die Producer-Heartbeat-Time ein Wert größer Null eingesetzt, so ist sie aktiv und unterbindet das Node-/ Life-Guarding (siehe Seite 39). Wird die Producer-Heartbeat-Time auf ‘0’ gesetzt, so wird das Senden des Heartbeat durch dieses Modul beendet.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert [Dez]	Datentyp	R/W
1017_h	0	*Producer heartbeat time*	0 … FFFF	0 ms	unsigned 16	rw

producer-heartbeat_time Zykluszeit des Heartbeat-Producers zum Senden des Heartbeat auf dem Node-Guarding-ID (siehe Object 100Eh).Die Consumer-Heartbeat-Time der überwachenden Module muss immer größer sein als die Producer-Heartbeat-Time dieses Heartbeat sendenden Moduls.

2.5.8.17 Identity Object (1018_h)

Index	Name	Datentyp	R/W	Default-Wert
1018_h	*Identity object*	unsigned 32	ro	-

Dieses Objekt enthält allgemeine Informationen zum Gateway.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert	Datentyp	R/W
1018_h	0	*no_of_entries*	4	4	unsigned 8	ro
	1	*Vendor_ID*	0…FFFFFFFF	0000 0064_h	unsigned 32	const
	2	*product_code*	0…FFFFFFFF	716459_d	unsigned 32	const
	3	*revision_number*	0…FFFFFFFF	x.y	unsigned 32	ro
	4	*serial_number*	0…FFFFFFFF	0	unsigned 32	ro

Bedeutung der Variablen

Vendor_ID Diese Variable enthält die Vendor-ID. Hier ist immer der Wert 00000064_h eingetragen.

product_code Hier ist die Artikelnummer des Produkts abgelegt. Die Hex.-Zahl entspricht dezimal der Artikelnummer.

revision_number Hier ist die Software-Version abgelegt. In den oberen zwei Bytes sind gemäß DS-301 die Revisionsnummern der wesentlichen (Major) Änderungen aufgeführt und in den unteren zwei Bytes die Revisionsnummern einfacher Korrekturen oder Änderungen (Minor).

2.5.8.18 Verify Configuration (1020_h)

Index	Name	Datentyp	R/W	Default-Wert
1020_h	*Verify configuration*	unsigned 32	rw	00_h

In diesem Objekt können Datum und Tageszeit der letzten Konfiguration abgelegt werden, um zu einem späteren Zeitpunkt prüfen zu können, ob die gespeicherte Konfiguration der erwarteten entspricht.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert [Dez]	Datentyp	R/W
1020_h	0	*no_of_entries*	2	2	unsigned 8	ro
	1	*configuration_date*	0…FFFFFFFF	0	unsigned 32	rw
	2	*configuration_time*	0…FFFFFFFF	0	unsigned 32	rw

Bedeutung der Variablen

configuration_date Datum der letzten Konfiguration des Moduls, angegeben in Tagen seit dem 01.01.1984.

configuration_time Zeit in ms seit Mitternacht am Tag der letzten Konfiguration.



2.5.8.19 Error Behaviour Object (1029_h)

Index	Name	Datentyp	R/W	Default-Wert
1029_h	*Error behaviour object*	unsigned 8	rw	00_h

Tritt ein Error-Event ein (z.B. Heartbeat-Error), so wechselt das Modul in den Zustand, der in der Variablen communication_error oder output_error definiert ist.

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default-Wert [Dez]	Datentyp	R/W
1020_h	0	*no_of_error_classes*	6	6	unsigned 8	ro
	1	*Communication_error*	0…2	0	unsigned 8	rw

Bedeutung der Variablen

no_of_error_classes Anzahl der Fehler-Klassen (hier immer ‘6’)

Communication_error Heartbeat/Lifeguard-Fehler und Bus off

Das Modul wechselt beim Auftreten eines Fehlers in den jeweils angegebenen Modus.

Wert der Variablen	Modus, in den das Modul im Fehlerfall wechselt
0	pre-operational (nur wenn der aktuelle Zustand = operational)
1	no state change
2	stopped

2.5.8.20 Receive PDO Communication Parameter ($140x_h$)

Mit diesen Objekten ‘Receive PDO Communication Parameter’ werden die Eigenschaften eines Empfangs-PDOs (Rx-PDO) definiert.

Index	Name	Datentyp	R/W
1400_h	*1. Receive PDO Parameter*	PDO CommPar	rw
1401_h	*2. Receive PDO Parameter*	PDO CommPar	rw
1402_h	*3. Receive PDO Parameter*	PDO CommPar	rw
1403_h	*4. Receive PDO Parameter*	PDO CommPar	rw

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default	Datentyp	R/W
1400_h	0	*no_of_entries*	2	2	unsigned 8	ro
	1	*COB-ID used by PDO1*	1…800007FF	200_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
1401_h	0	*no_of_entries*	2	2	unsigned 8	ro
	1	*COB-ID used by PDO2*	1…800007FF	300_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
1402_h	0	*no_of_entries*	2	2	unsigned 8	ro
	1	*COB-ID used by PDO3*	1…800007FF	400_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
1403_h	0	*no_of_entries*	2	2	unsigned 8	ro
	1	*COB-ID used by PDO4*	1…800007FF	500_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw

Es werden alle transmission types 0-240, 254 und 255 unterstützt.

2.5.8.21 Receive PDO Mapping Parameter (160x$_h$)

Mit den Objekten 'Receive PDO Mapping Parameter ' wird die Zuordnung der Empfangsdaten zu den Rx-PDOs definiert.

Index	Name	Datentyp	R/W
1600$_h$	*1. Receive PDO Mapping*	PDO Mapping	ro
1601$_h$	*2. Receive PDO Mapping*	PDO Mapping	ro
1602$_h$	*3. Receive PDO Mapping*	PDO Mapping	ro
1603$_h$	*4. Receive PDO Mapping*	PDO Mapping	ro

Die folgende Tabelle zeigt die Belegung der Receive PDO Mapping Parameter für die Default-Konfiguration:

Index	Sub-Index [Dez]	Beschreibung	Default [Hex]	Datentyp	R/W
1600$_h$	0	*no_of_entries*	1	unsigned 8	ro
	1	*1st_application_object*	62000108	unsigned 32	ro
	2	*2nd_application_object*	62000208	unsigned 32	ro
	3	*3rd_application_object*	62000308	unsigned 32	ro
	4	*4th_application_object*	62000408	unsigned 32	ro
	5	*5th_application_object*	62000508	unsigned 32	ro
	6	*6th_application_object*	62000608	unsigned 32	ro
	7	*7th_application_object*	62000708	unsigned 32	ro
	8	*8th_application_object*	62000808	unsigned 32	ro
1601$_h$	0	*no_of_entries*	1	unsigned 8	ro
	1	*1st_application_object*	62000908	unsigned 32	ro
	2	*2nd_application_object*	62000A08	unsigned 32	ro
	3	*3rd_application_object*	62000B08	unsigned 32	ro
	4	*4th_application_object*	62000C08	unsigned 32	ro
	5	*5th_application_object*	62000D08	unsigned 32	ro
	6	*6th_application_object*	62000E08	unsigned 32	ro
	7	*7th_application_object*	62000F08	unsigned 32	ro
	8	*8th_application_object*	62001008	unsigned 32	ro
1602$_h$	0	*no_of_entries*	1	unsigned 8	ro
	1	*1st_application_object*	62001108	unsigned 32	ro
	2	*2nd_application_object*	62001208	unsigned 32	ro
	3	*3rd_application_object*	62001308	unsigned 32	ro
	4	*4th_application_object*	62001408	unsigned 32	ro
	5	*5th_application_object*	62001508	unsigned 32	ro
	6	*6th_application_object*	62001608	unsigned 32	ro
	7	*7th_application_object*	62001708	unsigned 32	ro
	8	*8th_application_object*	62001808	unsigned 32	ro
1603$_h$	0	*no_of_entries*	1	unsigned 8	ro
	1	*1st_application_object*	62001908	unsigned 32	ro
	2	*2nd_application_object*	62001A08	unsigned 32	ro
	3	*3rd_application_object*	62001B08	unsigned 32	ro
	4	*4th_application_object*	62001C08	unsigned 32	ro
	5	*5th_application_object*	62001D08	unsigned 32	ro
	6	*6th_application_object*	62001E08	unsigned 32	ro
	7	*7th_application_object*	62001F08	unsigned 32	ro
	8	*8th_application_object*	62002008	unsigned 32	ro

2.5.8.22 Transmit PDO Communication Parameter ($180x_h$)

Mit diesem Objekt werden die Eigenschaften eines Sende-PDO1 definiert.

Index	Name	Datentyp	R/W
1800_h	*1. Transmit PDO Parameter*	PDO CommPar	ro
1801_h	*2. Transmit PDO Parameter*	PDO CommPar	ro
1802_h	*3. Transmit PDO Parameter*	PDO CommPar	ro
1803_h	*4. Transmit PDO Parameter*	PDO CommPar	ro

Index	Sub-Index [Dez]	Beschreibung	Wertebereich [Hex]	Default [Hex]	Datentyp	R/W
1800_h	0	*no_of_entries*	0…FF	5	unsigned 8	ro
	1	*COB-ID used by PDO1*	1…800007FF	180_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
	3	*inhibit time*	0…FFFF	0	unsigned 16	rw
	4	*reserved*	0…FF	0	unsigned 8	const
	5	*event timer*	0…FFFF	0	unsigned 16	rw
1801_h	0	*no_of_entries*	0…FF	5	unsigned 8	ro
	1	*COB-ID used by PDO2*	1…800007FF	280_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
	3	*inhibit time*	0…FFFF	0	unsigned 16	rw
	4	*reserved*	0…FF	0	unsigned 8	const
	5	*event timer*	0…FFFF	0	unsigned 16	rw
1802_h	0	*no_of_entries*	0…FF	5	unsigned 8	ro
	1	*COB-ID used by PDO3*	1…800007FF	380_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
	3	*inhibit time*	0…FFFF	0	unsigned 16	rw
	4	*reserved*	0…FF	0	unsigned 8	const
	5	*event timer*	0…FFFF	0	unsigned 16	rw
1803_h	0	*no_of_entries*	0…FF	5	unsigned 8	ro
	1	*COB-ID used by PDO4*	1…800007FF	480_h+Node-ID	unsigned 32	rw
	2	*transmission type*	0…FF	255_d	unsigned 8	rw
	3	*inhibit time*	0…FFFF	0	unsigned 16	rw
	4	*reserved*	0…FF	0	unsigned 8	const
	5	*event timer*	0…FFFF	0	unsigned 16	rw

Es werden die transmission types 0, 1-240, 252, 253, 254 und 255 unterstützt.

2.5.8.23 Transmit PDO Mapping Parameter ($1A0x_h$)

Mit den Objekten ‘Transmit PDO Mapping Parameter ’ wird die Zuordnung der Sendedaten zu den Tx-PDOs definiert.

Index	Name	Datentyp	R/W
$1A00_h$	*1. Receive PDO Mapping*	PDO Mapping	ro
$1A01_h$	*2. Receive PDO Mapping*	PDO Mapping	ro
$1A02_h$	*3. Receive PDO Mapping*	PDO Mapping	ro
$1A03_h$	*4. Receive PDO Mapping*	PDO Mapping	ro

Die folgende Tabelle zeigt die Belegung der Transmit PDO Mapping Parameter:

Index	Sub-Index	Beschreibung	Default [Hex]	Datentyp	R/W
$1A00_h$	0	*no_of_entries*	4	unsigned 32	ro
	1	*object_to_be_mapped_1*	60000108	unsigned 32	ro
	2	*object_to_be_mapped_2*	60000208	unsigned 32	ro
	3	*object_to_be_mapped_3*	60000308	unsigned 32	ro
	4	*object_to_be_mapped_4*	60000408	unsigned 32	ro
	5	*object_to_be_mapped_5*	60000508	unsigned 32	ro
	6	*object_to_be_mapped_6*	60000608	unsigned 32	ro
	7	*object_to_be_mapped_7*	60000708	unsigned 32	ro
	8	*object_to_be_mapped_8*	60000808	unsigned 32	ro
$1A01_h$	0	*no_of_entries*	4	unsigned 32	ro
	1	*object_to_be_mapped_1*	60000908	unsigned 32	ro
	2	*object_to_be_mapped_2*	60000A08	unsigned 32	ro
	3	*object_to_be_mapped_3*	60000B08	unsigned 32	ro
	4	*object_to_be_mapped_4*	60000C08	unsigned 32	ro
	5	*object_to_be_mapped_5*	60000D08	unsigned 32	ro
	6	*object_to_be_mapped_6*	60000E08	unsigned 32	ro
	7	*object_to_be_mapped_7*	60000F08	unsigned 32	ro
	8	*object_to_be_mapped_8*	60001008	unsigned 32	ro
$1A02_h$	0	*no_of_entries*	4	unsigned 32	ro
	1	*object_to_be_mapped_1*	60001108	unsigned 32	ro
	2	*object_to_be_mapped_2*	60001208	unsigned 32	ro
	3	*object_to_be_mapped_3*	60001308	unsigned 32	ro
	4	*object_to_be_mapped_4*	60001408	unsigned 32	ro
	5	*object_to_be_mapped_5*	60001508	unsigned 32	ro
	6	*object_to_be_mapped_6*	60001608	unsigned 32	ro
	7	*object_to_be_mapped_7*	60001708	unsigned 32	ro
	8	*object_to_be_mapped_8*	60001808	unsigned 32	ro
$1A03_h$	0	*no_of_entries*	4	unsigned 32	ro
	1	*object_to_be_mapped_1*	60001908	unsigned 32	ro
	2	*object_to_be_mapped_2*	60001A08	unsigned 32	ro
	3	*object_to_be_mapped_3*	60001B08	unsigned 32	ro
	4	*object_to_be_mapped_4*	60001C08	unsigned 32	ro
	5	*object_to_be_mapped_5*	60001D08	unsigned 32	ro
	6	*object_to_be_mapped_6*	60001E08	unsigned 32	ro
	7	*object_to_be_mapped_7*	60001F08	unsigned 32	ro
	8	*object_to_be_mapped_8*	60002008	unsigned 32	ro

2.5.9 Device Profile Area (6000_h … $9FFF_h$)

Index	Sub-Index	Beschreibung	Datentyp	R/W	Beispiel im Kapitel
6000_h	32	Modulstatus AN / AUS, (Byteweise) Sub-Index 01h = Knoten 1-8, Sub-Index 02h = Knoten 9-16 … Sub-Index 20h = Knoten 249-254 Binär 0 = AUS / Ausgelöst / Ext.-Aus Binär 1 = An	uint 8	ro	4.4.8.1
6020_h	128	Modulstatus AN / AUS, (Bitweise) Sub-Index 01h = Knoten 1, Sub-Index 02h = Knoten 2 … Sub-Index 80h = Knoten 128	Bool	ro	4.4.8.2
6021_h	126	Modulstatus AN / AUS, (Bitweise) Sub-Index 01h = Knoten 129, Sub-Index 02h = Knoten 130 … Sub-Index 7Eh = Knoten 254	Bool	ro	4.4.8.3
6200_h	32	Module Ein- / Ausschalten, (Byteweise) Sub-Index 01h = Knoten 1-8, Sub-Index 02h = Knoten 9-16 … Sub-Index 20h = Knoten 249-254 Binär 0 = AUS / Ausgelöst / Ext.-Aus Binär 1 = An	unit 8	rw	4.4.8.4
6220_h	128	Module Ein- / Ausschalten, (Bitweise) Sub-Index 01h = Knoten 1, Sub-Index 02h = Knoten 2 … Sub-Index 80h = Knoten 128	Bool	rw	4.4.8.5
6221_h	126	Module Ein- / Ausschalten, (Bitweise) Sub-Index 01h = Knoten 129, Sub-Index 02h = Knoten 130 … Sub-Index 7Eh = Knoten 254	Bool	rw	4.4.8.6

ro = nur lesen
rw = lesen, schreiben

2.5.9.1 Modulstatus AN / AUS (6000$_h$)

Index	Name	Datentyp	R/W
6000$_h$	*Modulstatus AN / AUS - Byteweise*	uint 8	ro

Der gelesene Wert gibt Byteweise den *Modulzustand AN / AUS* der angeschlossenen LOCC-Box-Net Module zurück.

Auslesen des Modulstatus

Der CANopen Master sendet unter dem Identifier ‘605$_h$’ (600$_h$ + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5$_h$).

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für je eine 8 Bit Gruppe (1 Byte).

LOCC-Box-Net Knotennummer 1 - 8 = 01$_h$
LOCC-Box-Net Knotennummer 9 - 16 = 02$_h$
LOCC-Box-Net Knotennummer 17 - 24 = 03$_h$
: :
LOCC-Box-Net Knotennummer 241 - 248 = 1F$_h$
LOCC-Box-Net Knotennummer 249 - 254 = 20$_h$

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605$_h$	0$_h$	8$_h$	40$_h$	00$_h$	60$_h$	01$_h$	00$_h$	00$_h$	00$_h$	00$_h$
			Read Request	Index=6000$_h$		Module / Knoten 1-8				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response (4F$_h$ = 1 Datenbyte) unter dem Identifier ‘585$_h$’ (580$_h$ + Node-ID) mit dem Status der Module mit der Knotennummer 1 - 8.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
585$_h$	0$_h$	8$_h$	4F$_h$	00$_h$	60$_h$	01$_h$	03$_h$	00$_h$	00$_h$	00$_h$
			Read Response	Index=6000$_h$		Module / Knoten 1-8	gelesener Wert: 00 00 00 03$_h$			

Der gelesene Wert wird in einen Binärwert gewandelt.

Beispiel 1:
Datenbyte 03$_h$ = binär: 0 0 0 0 0 0 1 1
Sub-Index 01$_h$ = Knoten-Nr.: 8 7 6 5 4 3 2 1
→ Module 1 und 2 sind AN
Module 3 - 8 sind AUS

Beispiel 2:
Datenbyte 20$_h$ = binär: X X 1 0 0 0 0 0
Sub-Index 20$_h$ = Knoten-Nr.: 256 255 254 253 252 251 250 249
→ Modul 254 ist AN
Modul 249 – 253 sind AUS

Bemerkung: Pro Gateway können maximal 254 LOCC-Box-Net Module verwaltet werden!

2.5.9.2 Modulstatus AN / AUS (6020$_h$)

Index	Name	Datentyp	R/W
6020$_h$	*Modulstatus AN / AUS – Bit, Knoten 1 - 128*	uint 8	ro

Der gelesene Wert gibt den *Modulzustand AN / AUS* des ausgewählten Modules zurück.

Auslesen des Modulstatus

Der CANopen Master sendet unter dem Identifier ‘605$_h$’ (600$_h$ + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5$_h$).

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für die untergeordneten Module 1 - 128

LOCC-Box-Net Knotennummer 1 = 01$_h$
LOCC-Box-Net Knotennummer 2 = 02$_h$
LOCC-Box-Net Knotennummer 3 = 03$_h$
: :
LOCC-Box-Net Knotennummer 127 = 7F$_h$
LOCC-Box-Net Knotennummer 128 = 80$_h$

Im Beispiel wird das Modul Nr. 8 abgefragt.

→ **Datenbyte (LSB):** Das Datenbyte ist wie folgt zu lesen:
01$_h$ = Modul ist eingeschaltet
00$_h$ = Modul ist ausgeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
605$_h$	0$_h$	8$_h$	40$_h$	20$_h$	60$_h$	08$_h$	00$_h$	00$_h$	00$_h$	00$_h$
			Read Request	Index=6020$_h$		Module / Knoten 8				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response (4F$_h$ = 1 Datenbyte) unter dem Identifier ‘585$_h$’ (580$_h$ + Node-ID) mit dem Status des Modules mit der Knotennummer 8.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
585$_h$	0$_h$	8$_h$	4F$_h$	20$_h$	60$_h$	08$_h$	01$_h$	00$_h$	00$_h$	00$_h$
			Read Response	Index=6020$_h$		Module / Knoten 8	gelesener Wert: 00 00 00 01$_h$			

Beispiel 1:
Sub-Index = 08$_h$ / Datenbyte = 01$_h$ → Modul mit der Knotennummer 8 ist eingeschaltet!

Beispiel 2:
Sub-Index = 08$_h$ / Datenbyte = 00$_h$ → Modul mit der Knotennummer 8 ist ausgeschaltet!



2.5.9.3 Modulstatus AN / AUS (6021_h)

Index	Name	Datentyp	R/W
6021_h	*Modulstatus AN / AUS – Bit, Knoten 129 - 254*	uint 8	ro

Der gelesene Wert gibt den *Modulzustand AN / AUS* des ausgewählten Modules zurück.

Auslesen des Modulstatus

Der CANopen Master sendet unter dem Identifier '605_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5_h).

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für die untergeordneten Module 129 - 254

LOCC-Box-Net Knotennummer 129 = 81_h
LOCC-Box-Net Knotennummer 130 = 82_h
LOCC-Box-Net Knotennummer 131 = 83_h
: :
LOCC-Box-Net Knotennummer 253 = FD_h
LOCC-Box-Net Knotennummer 254 = FE_h

Im Beispiel wird das Modul Nr. 130 abgefragt.

→ **Datenbyte (LSB):** Das Datenbyte ist wie folgt zu lesen:
01_h = Modul ist eingeschaltet
00_h = Modul ist ausgeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	40_h	20_h	60_h	82_h	00_h	00_h	00_h	00_h
			Read Request	Index=6020_h		Module / Knoten 130				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response ($4F_h$ = 1 Datenbyte) unter dem Identifier '585_h' (580_h + Node-ID) mit dem Status des Modules mit der Knotennummer 130.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
585_h	0_h	8_h	$4F_h$	20_h	60_h	82_h	01_h	00_h	00_h	00_h
			Read Response	Index=6020_h		Module / Knoten 130	gelesener Wert: 00 00 00 01_h			

Beispiel 1:
Sub-Index = 82_h / Datenbyte = 01_h → Modul mit der Knotennummer 130 ist eingeschaltet!

Beispiel 2:
Sub-Index = 82_h / Datenbyte = 00_h → Modul mit der Knotennummer 130 ist ausgeschaltet!

2.5.9.4 Module AN / AUS schalten - (6200$_h$)

Index	Name	Datentyp	R/W
6200$_h$	*Module AN / AUS schalten – Byteweise*	uint 8	rw

Der Wert (niederwertigstes Byte) setzt Byteweise ausgewählten LOCC-Box-Net Module in den Zustand AN oder AUS.

Module AN / AUS schalten

Der CANopen Master sendet unter dem Identifier '605$_h$' (600$_h$ + Node-ID) den Write Request an das Gateway mit der Nr. 5 (Node-ID=5$_h$).

→ **Befehlscode**: Entspricht 2F$_h$ - Write Request für 1 Datenbyte (siehe Kapitel 4.4.3.1)

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für je eine 8 Bit Gruppe (1 Byte).

LOCC-Box-Net Knotennummer 1 - 8 = 01$_h$
LOCC-Box-Net Knotennummer 9 - 16 = 02$_h$
LOCC-Box-Net Knotennummer 17 - 24 = 03$_h$
: :
LOCC-Box-Net Knotennummer 241 - 248 = 1F$_h$
LOCC-Box-Net Knotennummer 249 - 254 = 20$_h$

→ **Datenbyte (LSB):** Das Datenbyte ist wie folgt zu schreiben:
1 = Modul wird eingeschaltet
0 = Modul wird ausgeschaltet

Beispiel 1:
Datenbyte 03$_h$ = binär: 0 0 0 0 0 0 1 1
Sub-Index 01$_h$ = Knoten-Nr.: 8 7 6 5 4 3 2 1
→ Module 1 und 2 = AN
Module 3 - 8 = AUS

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
605$_h$	0$_h$	8$_h$	2F$_h$	00$_h$	62$_h$	01$_h$	03$_h$	00$_h$	00$_h$	00$_h$
			Write Request	Index=6200$_h$		Module / Knoten 1-8	zu schreibender Wert			

Beispiel 2:
Datenbyte 0F$_h$ = binär: X X 0 0 1 1 1 1
Sub-Index 20$_h$ = Knoten-Nr.: 256 255 254 253 252 251 250 249
→ Modul 249 - 252 = AN
Modul 253 – 254 = Aus

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
605$_h$	0$_h$	8$_h$	2F$_h$	00$_h$	62$_h$	20$_h$	0F$_h$	00$_h$	00$_h$	00$_h$
			Write Request	Index=6200$_h$		Module / Knoten 249-254	zu schreibender Wert			

Das Gateway-Modul Nr. 5 antwortet dem Master mit einer Write Response (60$_h$) unter dem Identifier '585$_h$' (580$_h$ + Node-ID). Die Antwort enthält keine Daten.



2.5.9.5 Modul AN / AUS schalten - (6220_h)

Index	Name	Datentyp	R/W
6220_h	*Modulstatus AN / AUS – Bit, Knoten 1 - 128*	uint 8	rw

Der Wert setzt das ausgewählten LOCC-Box-Net Modul in den Zustand AN oder AUS.

Modul AN / AUS schalten

Der CANopen Master sendet unter dem Identifier ‘605_h’ (600_h + Node-ID) den Write Request an das Gateway mit der Nr. 5 (Node-ID=5_h).

→ **Befehlscode**: Entspricht $2F_h$ - Write Request für 1 Datenbyte (siehe Kapitel 4.4.3.1)

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für die untergeordneten Module 1 – 128

LOCC-Box-Net Knotennummer 1 = 01_h
LOCC-Box-Net Knotennummer 2 = 02_h
LOCC-Box-Net Knotennummer 3 = 03_h
: :
LOCC-Box-Net Knotennummer 127 = $7F_h$
LOCC-Box-Net Knotennummer 128 = 80_h

→ **Datenbyte (LSB):** Das Datenbyte ist wie folgt zu schreiben:
01_h = Modul wird eingeschaltet
00_h = Modul wird ausgeschaltet

Beispiel 1:
Sub-Index = 01_h / Datenbyte = 01_h → Modul mit der Knotennummer 1 wird eingeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	$2F_h$	20_h	62_h	01_h	01_h	00_h	00_h	00_h
			Write Request	Index=6220_h		Modul / Knoten 1	zu schreibender Wert			

Beispiel 2:
Sub-Index = 14_h / Datenbyte = 00_h → Modul mit der Knotennummer 20 wird ausgeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	$2F_h$	20_h	62_h	14_h	00_h	00_h	00_h	00_h
			Write Request	Index=6220_h		Modul / Knoten 20	zu schreibender Wert			

Das Gateway-Modul Nr. 5 antwortet dem Master mit einer Write Response (60_h) unter dem Identifier ‘585_h’ (580_h + Node-ID). Die Antwort enthält keine Daten.
Wird ein bereits eingeschaltes Modul erneut eingeschaltet oder ein bereits ausgeschaltes Modul erneut ausgeschaltet, so erhält man ein Error Response (80_h).



2.5.9.6 Modul AN / AUS schalten - (6221_h)

Index	Name	Datentyp	R/W
6221_h	*Modulstatus AN / AUS – Bit, Knoten 129 - 254*	uint 8	rw

Der Wert setzt das ausgewählten LOCC-Box-Net Modul in den Zustand AN oder AUS.

Modul AN / AUS schalten

Der CANopen Master sendet unter dem Identifier ‘605_h’ (600_h + Node-ID) den Write Request an das Gateway mit der Nr. 5 (Node-ID=5_h).

→ **Befehlscode**: Entspricht $2F_h$ - Write Request für 1 Datenbyte (siehe Kapitel 4.4.3.1)

→ **Sub-Index:** Der Sub-Index gilt aufsteigend für die untergeordneten Module 129 – 254_d

LOCC-Box-Net Knotennummer 129 = 01_h
LOCC-Box-Net Knotennummer 130 = 02_h
LOCC-Box-Net Knotennummer 131 = 03_h
: :
LOCC-Box-Net Knotennummer 253 = $7D_h$
LOCC-Box-Net Knotennummer 254 = $7E_h$

→ **Datenbyte (LSB):** Das Datenbyte ist wie folgt zu schreiben:
01_h = Modul wird eingeschaltet
00_h = Modul wird ausgeschaltet

Beispiel 1:
Sub-Index = 03_h / Datenbyte = 01_h → Modul mit der Knotennummer 131 wird eingeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	$2F_h$	21_h	62_h	03_h	01_h	00_h	00_h	00_h
			Write Request	Index=6221_h		Modul / Knoten 131	zu schreibender Wert			

Beispiel 2:
Sub-Index = $7D_h$ / Datenbyte = 00_h → Modul mit der Knotennummer 253 wird ausgeschaltet

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	$2F_h$	21_h	62_h	$7D_h$	00_h	00_h	00_h	00_h
			Write Request	Index=6221_h		Modul / Knoten 253	zu schreibender Wert			

Das Gateway-Modul Nr. 5 antwortet dem Master mit einer Write Response (60_h) unter dem Identifier ‘585_h’ (580_h + Node-ID). Die Antwort enthält keine Daten.
Wird ein bereits eingeschaltes Modul erneut eingeschaltet oder ein bereits ausgeschaltes Modul erneut ausgeschaltet, so erhält man ein Error Response (80_h).

2.5.10 Manufacturer Specific Profile Area (2000_h … $5FFF_h$)

Index	Sub-Index (max.)[Dez]	Beschreibung	Datentyp	R/W	Beispiel im Kapitel
2000_h	254	Modultyp von Knoten 1 – 254	uint 8	ro	4.4.9.1
2010_h	254	Modulstatus von Knoten 1 – 254	uint 8	ro	4.4.9.2
2011_h	254	Modulkonfiguration (Drehschalter)	uint 8	ro	4.4.9.3
$2100\text{-}210A_h$	254	Analogwerte Strom / Spannung	uint 16	ro	
2100_h	254	Ausgangsspannung			4.4.9.4
2101_h	254	Eingangsspannung			4.4.9.5
2104_h	254	Strommessung			4.4.9.6
$210A_h$	254	Kennlinieneinstellungen			4.4.9.7
$2200\text{-}2206_h$	254	Gerätedaten und Zählerstände	uint 32	ro	
2200_h	254	Softwareversion			4.4.9.8
2201_h	254	Seriennummer			4.4.9.9
2202_h	254	LOCC-Box Zähler “Betriebsspg.An“			4.4.9.10
2203_h	254	LOCC-Box Zähler “Betriebsstunden (h)“			4.4.9.11
2204_h	254	LOCC-Box Zähler “Betriebsstunden AN (h)“			4.4.9.12
2205_h	254	LOCC-Box Zähler “Ausgelöst“ (siehe S. 23)			4.4.9.13
2206_h	254	LOCC-Box Zähler “Einschalten“ (siehe S. 24)			4.4.9.14

Sub-Index = Knotennummer der angeschlossenen LOCC-Box
ro = nur lesen
rw = lesen und schreiben



2.5.10.1 Modul Typ (2000$_h$)

Index	Name	Datentyp	R/W
2000$_h$	*Modul typ*	uint 8	ro

Der Wert des *Modul Typ* interpretiert die Modul Version: 1 = 716410

Auslesen des Modul Typ

Der CANopen Master sendet unter dem Identifier ‘605$_h$’ (600$_h$ + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5$_h$).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 9.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
605$_h$	0$_h$	8$_h$	40$_h$	00$_h$	20$_h$	09$_h$	00$_h$	00$_h$	00$_h$	00$_h$
			Read Request	Index=2000$_h$		Modul / Knoten 9				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response (4F$_h$ = 1 Datenbyte) unter dem Identifier ‘585$_h$’ (580$_h$ + Node-ID) mit dem Wert des Modul mit der Knotennummer 9:

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
585$_h$	0$_h$	8$_h$	4F$_h$	00$_h$	20$_h$	09$_h$	01$_h$	00$_h$	00$_h$	00$_h$
			Read Response (1Byte)	Index=2000$_h$		Modul / Knoten 9	gelesener Wert: 00 00 00 01$_h$			

Der gelesene Wert wird in einen Dezimalwert gewandelt.

Beispiel:

00 00 00 01$_h$ = 1

Entspricht dem Modul Typ: 1 → 716410

2.5.10.2 Modulzustand (2010_h)

Index	Name	Datentyp	R/W
2010_h	*Modulzustand*	uint 8	ro

Der ausgelesene Wert gibt den *Modulzustand* an.

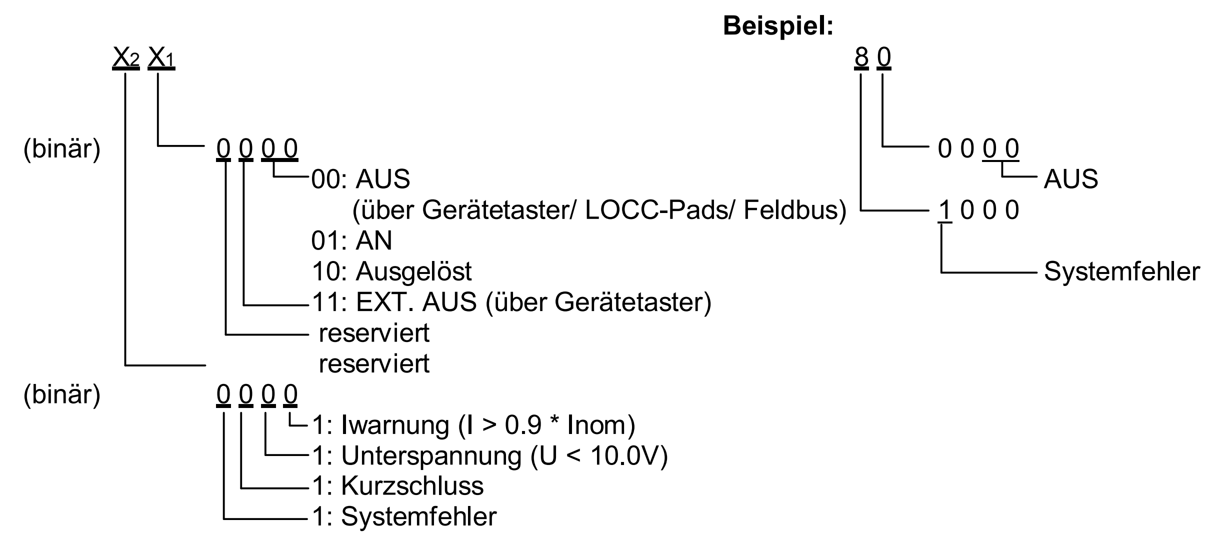
Auslesen des Modulzustandes
Der CANopen Master sendet unter dem Identifier '605_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 9.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	40_h	10_h	20_h	09_h	00_h	00_h	00_h	00_h
			Read Request	Index=2010_h		Modul / Knoten 9				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response ($4F_h$ = 1 Datenbyte) unter dem Identifier '585_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 9.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
585_h	0_h	8_h	$4F_h$	10_h	20_h	09_h	80_h	00_h	00_h	00_h
			Read Response (1Byte)	Index=2010_h		Modul / Knoten 9	gelesener Wert: 00 00 00 80_h			

Der gelesene Wert wird in einen Binärwert gewandelt.



Ergebnis ist: Modul ist AUS mit Hilfe des Gerätetaster/ LOCC-Pads oder Feldbus und signalisiert einen Systemfehler.

2.5.10.3 Modulkonfiguration (2011_h)

Index	Name	Datentyp	R/W
2011_h	*Modulkonfiguration*	uint 8	ro

Der Wert der *Modulkonfiguration* gibt die Einstellung des Strombereiches bzw. der Charakteristik (Drehschalter) zurück.

Auslesen der Modulkonfiguration
Der CANopen Master sendet unter dem Identifier '605_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	40_h	11_h	20_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=2011_h		Modul / Knoten 1				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response ($4F_h$ = 1 Datenbyte) unter dem Identifier '585_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
585_h	0_h	8_h	$4F_h$	11_h	20_h	01_h	15_h	00_h	00_h	00_h
			Read Response (1Byte)	Index=2011_h		Modul / Knoten 1	gelesener Wert: 00 00 00 15_h			

Der gelesene Wert wird in einen Dezimalwert gewandelt und mit 1 addiert.

Y_2 Y_1
└ = Dezimalwert 0-9 + 1 = Strombereich 1-10A
└── = Dezimalwert 0-9 + 1 = Charakteristik 1-10

Beispiel:

1 5
└ = dezimal = 5 + 1 = Strombereich 6A
└── = dezimal = 1 + 1 = Charakteristik 2

2.5.10.4 Ausgangsspannung (2100_h)

Index	Name	Datentyp	R/W
2100_h	*Ausgangsspannung*	uint 16	ro

Der Wert enthält den Betrag der anliegenden *Ausgangspannung*.

Auslesen der Ausgangsspannung
Der CANopen Master sendet unter dem Identifier '$60A_h$' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 10 (Node-ID=10_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
$60A_h$	0_h	8_h	40_h	00_h	21_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=2100_h		Modul / Knoten 1				

Das Gateway-Modul Nr. 10 antwortet dem Master anhand der Read Response ($4B_h$ = 2 Datenbytes) unter dem Identifier '$58A_h$' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
$58A_h$	0_h	8_h	$4B_h$	00_h	21_h	01_h	$9C_h$	02_h	00_h	00_h
			Read Response (2Byte)	Index=2100_h		Modul / Knoten 1	gelesener Wert: 00 00 02 $9C_h$			

Der gelesene Wert wird in einen Dezimalwert gewandelt. Der maximale Messwert ist 1024 und entspricht 39V. Hieraus ergibt sich folgende Verhältnisgleichung.

$$\text{Ausgangsspannung} = \frac{\textit{Dezimalwert} \times 39V}{1024}$$

Beispiel:

00 00 02 $9C_h$ = dezimal = 668

$$\text{Ausgangsspannung} = \frac{\textit{668} \times 39V}{1024} = \underline{25{,}44V}$$

2.5.10.5 Eingangsspannung (2101_h)

Index	Name	Datentyp	R/W
2101_h	*Eingangsspannung*	uint 16	ro

Der Wert enthält den Betrag der anliegenden *Eingangsspannung*.

Auslesen der Eingangsspannung
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	01_h	21_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=2101_h		Modul / Knoten 1				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response ($4B_h$ = 2 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	$4B_h$	01_h	21_h	01_h	98_h	02_h	00_h	00_h
			Read Response (2Byte)	Index=2101_h		Modul / Knoten 1	gelesener Wert: 00 00 02 98_h			

Der gelesene Wert wird in einen Dezimalwert gewandelt. Der maximale Messwert ist 1024 und entspricht 39V. Hieraus ergibt sich folgende Verhältnisgleichung.

$$\text{Eingangsspannung} = \frac{\textit{Dezimalwert} \times 39V}{1024}$$

Beispiel:

00 00 02 98_h = dezimal = 664

$$\text{Eingangsspannung} = \frac{664 \times 39V}{1024} = \underline{\underline{25{,}29V}}$$

2.5.10.6 Strommessung (2104_h)

Index	Name	Datentyp	R/W
2104_h	*Strommessung*	uint 16	ro

Der Wert enthält den Betrag des fließenden *Stromes*.

Auslesen - Strommessung
Der CANopen Master sendet unter dem Identifier '$60A_h$' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 10 (Node-ID=10_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
$60A_h$	0_h	8_h	40_h	04_h	21_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=2104_h		Modul / Knoten 1				

Das Gateway-Modul Nr. 10 antwortet dem Master anhand der Read Response ($4B_h$ = 2 Datenbytes) unter dem Identifier '$58A_h$' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
$58A_h$	0_h	8_h	$4B_h$	04_h	21_h	01_h	$1F_h$	00_h	00_h	00_h
			Read Response (2Byte)	Index=2104_h		Modul / Knoten 1	gelesener Wert: 00 00 00 $1F_h$			

Der gelesene Wert wird in einen Dezimalwert gewandelt. Der maximale Messwert ist 1024 und entspricht 32,75A. Hieraus ergibt sich folgende Verhältnisgleichung.

$$\text{Strom} = \frac{\textit{Dezimalwert} \times 32{,}75\text{A}}{1024}$$

Beispiel:

00 00 00 $1F_h$ = dezimal = 31

$$\text{Strom} = \frac{\textit{31} \times 32{,}75\text{A}}{1024} = \underline{\underline{0{,}99\text{A}}}$$

2.5.10.7 Kennlinieneinstellung ($210A_h$)

Index	Name	Datentyp	R/W
$210A_h$	*Kennlinieneinstellung*	uint 16	ro

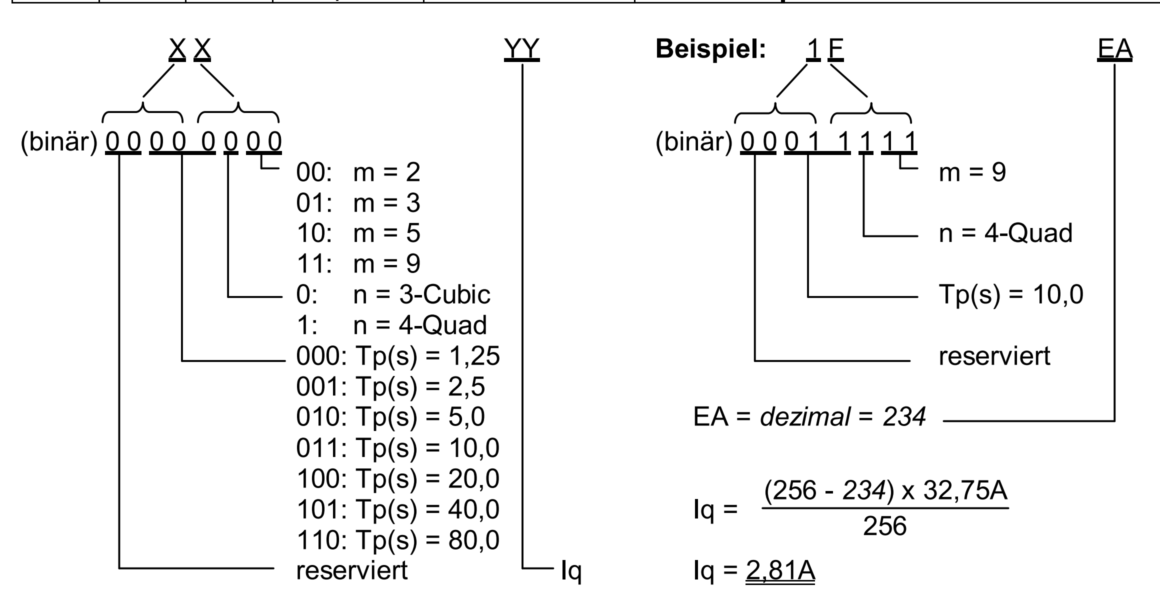
Dieses Objekt liefert die aktuellen Parameter der eingestellten Kennlinie (Charakteristik).

Auslesen - Kennlinieneinstellungen
Der CANopen Master sendet unter dem Identifier '605_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 5 (Node-ID=5_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
605_h	0_h	8_h	40_h	$0A_h$	21_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=$210A_h$		Modul / Knoten 1				

Das Gateway-Modul Nr. 5 antwortet dem Master anhand der Read Response ($4B_h$ = 2 Datenbytes) unter dem Identifier '585_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
585_h	0_h	8_h	$4B_h$	$0A_h$	21_h	01_h	EA_h	$1F_h$	00_h	00_h
			Read Response	Index=$210A_h$		Modul / Knoten 1	gelesener Wert: 00 00 1F EA_h (xx yy)			



Umwandlung in *Dezimalwert.* Der maximale Messwert ist 256 und entspricht 32,75A.

Hieraus ergibt sich folgender Dreisatz: $Iq = \frac{(256 - \text{Dezimalwert}) \times 32{,}75A}{256}$

2.5.10.8 Softwareversion (2200_h)

Index	Name	Datentyp	R/W
2200_h	*Softwareversion*	uint 32	ro

Dieses Objekt liefert die *Softwareversion* der LOCC-Box.

Auslesen - Softwareversion
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	00_h	22_h	01_h	00_h	00_h	00_h	00_h
			Read Request	Index=2200_h		Modul / Knoten 1				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 1.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	00_h	22_h	01_h	12_h	00_h	00_h	00_h
			Read Response (4Byte)	Index=2200_h		Modul / Knoten 1	gelesener Wert: 00 00 00 12_h			

Der gelesene Wert wird als Hex-Wert gelesen und mit einer Kommastelle von rechts versehen.

Beispiel:

00 00 00 12_h = 1.2

2.5.10.9 Seriennummer (2201_h)

Index	Name	Datentyp	R/W
2201_h	*Seriennummer*	uint 32	ro

Dieses Objekt liefert die *Seriennummer* der LOCC-Box.

Auslesen - Seriennummer
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	01_h	22_h	02_h	00_h	00_h	00_h	00_h
			Read Request	Index=2201_h		Modul / Knoten 2				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	01_h	22_h	02_h	EF_h	$E1_h$	01_h	00_h
			Read Response (4Byte)	Index=2201_h		Modul / Knoten 2	gelesener Wert: 00 01 E1 EF_h			

Der gelesene Wert wird in einen Dezimalwert gewandelt.

Beispiel:

00 01 E1 EF_h = dezimal = 123375

2.5.10.10 LOCC-Box Zähler "Betriebsspannung AN" (2202_h)

Index	Name	Datentyp	R/W
2202_h	*LOCC-Box Zähler "Betriebsspannung AN"*	uint 32	ro

Dieses Objekt liefert den *Zählerstand* wie oft das Modul an die Versorgungsspannung angeschlossen wurde.

Auslesen - Betriebsspannung AN
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	02_h	22_h	02_h	00_h	00_h	00_h	00_h
			Read Request	Index=2202_h		Modul / Knoten 2				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	02_h	22_h	02_h	$0C_h$	01_h	00_h	00_h
			Read Response (4Byte)	Index=2202_h		Modul / Knoten 2	gelesener Wert: 00 00 01 $0C_h$			

Der gelesene Wert wird in einen Dezimalwert gewandelt.

Beispiel:

00 00 01 $0C_h$ = dezimal = 268



2.5.10.11 LOCC-Box Zähler “Betriebsstunden (h)“ (2203h)

Index	Name	Datentyp	R/W
2203h	*LOCC-Box Zähler “Betriebsstunden (h)“*	uint 32	ro

Dieses Objekt liefert die Anzahl der Betriebsstunden im ½ Stunden Takt, d.h. wie lange die LOCC-Box an die Versorgungsspannung angeschlossen war.

Auslesen – Betriebsstunden (h)
Der CANopen Master sendet unter dem Identifier ‘608h’ (600h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608h	0h	8h	40h	03h	22h	02h	00h	00h	00h	00h
			Read Request	Index=2203h		Modul / Knoten 2				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43h = 4 Datenbytes) unter dem Identifier ‘588h’ (580h + Node-ID) mit dem Wert des Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588h	0h	8h	43h	03h	22h	02h	60h	01h	00h	00h
			Read Response (4Byte)	Index=2203h		Modul / Knoten 2	gelesener Wert: 00 00 01 60h			

Der gelesene Wert wird in einen Dezimalwert gewandelt und durch 2 geteilt.

Beispiel:

00 00 01 60h = dezimal / 2 = 176h



2.5.10.12 LOCC-Box Zähler "Betriebsstunden AN (h)" (2204_h)

Index	Name	Datentyp	R/W
2204_h	*LOCC-Box Zähler "Betriebsstunden AN (h)"*	uint 32	ro

Dieses Objekt liefert die Anzahl der Betriebsstunden AN im ½ Stunden Takt, d.h. wie lange die LOCC-Box eingeschaltet war und den Verbraucher versorgt hat.

Auslesen – Betriebsstunden AN(h)
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	04_h	22_h	02_h	00_h	00_h	00_h	00_h
			Read Request	Index=2204_h		Modul / Knoten 2				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	04_h	22_h	02_h	FB_h	08_h	00_h	00_h
			Read Response (4Byte)	Index=2204_h		Modul / Knoten 2	gelesener Wert: 00 00 08 FB_h			

Der gelesene Wert wird in einen Dezimalwert gewandelt und durch 2 geteilt.

Beispiel:

00 00 08 FB_h = dezimal / 2 = 1149,5h



2.5.10.13 LOCC-Box Zähler “Ausgelöst“ (2205_h)

Index	Name	Datentyp	R/W
2205_h	*LOCC-Box Zähler “Ausgelöst“*	uint 32	ro

Dieses Objekt liefert die Anzahl wie oft die LOCC-Box durch Überlast oder Kurzschluss ausgelöst hat.

Auslesen – Ausgelöst
Der CANopen Master sendet unter dem Identifier ‘608_h’ (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	05_h	22_h	02_h	00_h	00_h	00_h	00_h
			Read Request	Index=2205_h		Modul / Knoten 2				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier ‘588_h’ (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 2.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	05_h	22_h	02_h	28_h	00_h	00_h	00_h
			Read Response (4Byte)	Index=2205_h		Modul / Knoten 2	gelesener Wert: 00 00 00 28_h			

Der gelesene Wert wird in einen Dezimalwert gewandelt.

Beispiel:

00 00 00 28_h = dezimal = 40

2.5.10.14 LOCC-Box Zähler "Einschalten" (2206_h)

Index	Name	Datentyp	R/W
2206_h	*LOCC-Box Zähler "Einschalten"*	uint 32	ro

Dieses Objekt liefert die Anzahl wie oft die LOCC-Box eingeschaltet wurde.

Auslesen – Einschalten
Der CANopen Master sendet unter dem Identifier '608_h' (600_h + Node-ID) den Read Request an das Gateway mit der Nr. 8 (Node-ID=8_h).
→ Ausgelesen wird das untergeordnete Modul mit der Knotennummer 3.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
608_h	0_h	8_h	40_h	06_h	22_h	03_h	00_h	00_h	00_h	00_h
			Read Request	Index=2206_h		Modul / Knoten 3				

Das Gateway-Modul Nr. 8 antwortet dem Master anhand der Read Response (43_h = 4 Datenbytes) unter dem Identifier '588_h' (580_h + Node-ID) mit dem Wert des Modul mit der Knotennummer 3.

ID	RTR	LEN	Daten							
			1	2	3	4	5	6	7	8
			Befehl-code	Index low	Index high	Sub-Index	LSB	…	…	MSB
588_h	0_h	8_h	43_h	06_h	22_h	03_h	$2C_h$	00_h	00_h	00_h
			Read Response (4Byte)	Index=2206_h		Modul / Knoten 3	gelesener Wert: 00 00 00 $2C_h$			

Der gelesene Wert wird in einen Dezimalwert gewandelt.

Beispiel:

00 00 00 $2C_h$ = dezimal = 44



2.6 Technische Daten

Allgemein Daten	
Nennspannung	DC 12/24V
Arbeitsspannungsbereich	DC 10 – 32V
Nennstrom	max. 50mA
Verpolungsschutz	ja
Anschlussart	Federzugklemmen
Gehäusematerial	PA 6.6 (UL 94 V0; NFF I2,F2)
Montage	aufrastbar auf TS 35 (gemäß EN 50022)
Schutzart	IP 20
Einbaulage	beliebig
Anschlusstechnik	Federzug Anschluss 0,25mm² – 2,5mm² alle Leiterarten bis 2,5mm² ohne Aderendhülse bis 1,5mm² mit Aderendhülse
USB	USB 2.0 Full-Speed (12 Mbit/s)
UART (RS232)	Baudrate 600 - 115200 bit/s
CANopen	Baudrate 10 – 1000 kbit/s
Arbeitstemperaturbereich	-20°C bis +60°C
Lagertemperaturbereich	-40°C bis +85°C
Maße (BxHxT)	6,2 x 90 x 115,5mm
Gewicht	0,06 kg
Zulassungen	CE
Normen	EN 60950-1; EN61131-1,2; EN 60947-4-1; EN 50081

LOCC-BUS	
Zugriffsverfahren	Single-Master - Multiple Slave
Bustechnologie	Linie
Physikalische Ebene	1-wire
Teilnehmer	typisch 40, max. 84
Buslänge	typisch 10m, maximal 40m
Übertragungsrate	9600 Baud
Datenrate	8 Bit + feste Parität
Übertragungs-Protokoll	Modifiziertes Multidrop

3 Austausch LOCC-Box-Net ohne LOCC-Pads

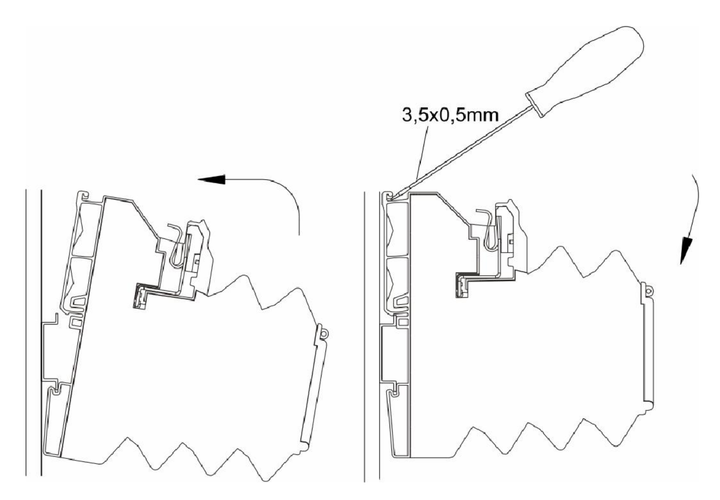
Der Austausch einer LOCC-Box-Net in einer bestehenden Konfiguration ist ohne Verwendung der Software LOCC-Pads möglich.

Voraussetzung ist:

- Es kann immer nur ein Modul getauscht werden.
- Das neue Modul muss im Auslieferungszustand sein, d.h. es muss die Knotennummer 0 haben.
- Es muss eine CANopen - Kommunikation bestehen.

Ablauf:

1. CANopen Kommunikation starten.
2. Brückungskämme entfernen und den Schiebekontakt am Anschluss 7 zurück schieben.
3. Module entfernen – siehe Bild



4. Drehschalter von Stromwert (I) und Charakteristik (C) am neuen Modul einstellen (siehe altes)
5. Modul aufrasten – siehe Bild
6. Schiebekontakt schließen und Brückungskämme wieder einsetzen.
7. Die neue LOCC-Box blinkt und muss durch betätigen des Gerätetasters bestätigt werden.

   Erfolgt keine Bestätigung innerhalb einer Minute so wird keine Knotennummer vergeben. Erst nach dem Ab- bzw. Zuschalten der Betriebsspannung ist die erneute Knotennummernvergabe der LOCC-Box-Net möglich.



Während dieser Zeit ist keine Kommunikation möglich!

8. Die getauschte LOCC-Box mit Hilfe des Gerätetasters aus- und wieder einschalten. Dies ist notwendig, um die geänderten Einstellungen von Strom und Charakteristik sicher zu übernehmen.

4 Firmware Update

4.1 Einleitung

Unter Berücksichtigung der Weiterentwicklung der Module aus der LOCC-Box-Net Familie sind Updates der Firmware im Gateway nicht ausgeschlossen.

4.2 Download

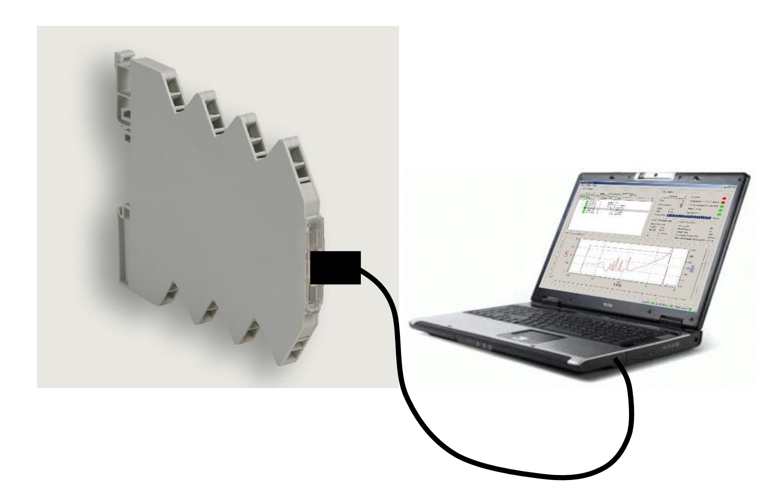
Für das Update wird das aktuelle Softwarepaket "LOCC-Pads" benötigt. Dieses steht als kostenloser Download, für registrierte Benutzer, unter http://www.luetze.de/downloads/software-interface/ zur Verfügung.
Laden Sie sich die Datei "LOCC-Pads_x.x.x.x.zip" herunter und speichern Sie diese in ein beliebiges Verzeichnis. Entpacken Sie die Datei, diese können Sie nach dem Update wieder löschen.

Im Ordner "Gateway Firmware" / „CANopen" ist die Firmware für das Gateway und im Ordner "EDS" die CANopen Slave Information enthalten. Bitte ersetzen Sie die EDS Datei in ihrer Steuerung.

4.3 Installation

1. Verbinden Sie den USB-Anschluss des Gateways über das mitgelieferte USB-Kabel mit dem PC (Laptop). Siehe Bild!

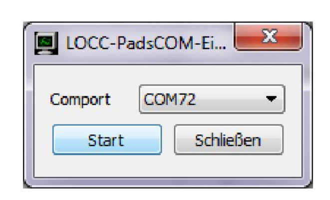
2. Versorgen Sie das Gateway, mit DC 12/24V Betriebsspannung.



4.4 Update

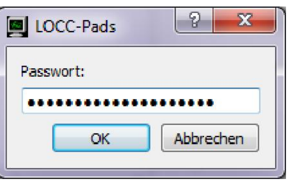
Gehen Sie beim Update wie folgt vor:

1. Starten Sie die Software LOCC-Pads durch Doppelklick auf das LOCC-Pads-Symbol auf dem Desktop oder führen Sie es über das Menü **Start / Programme / LOCC-Pads / LOCC-Pads** aus.



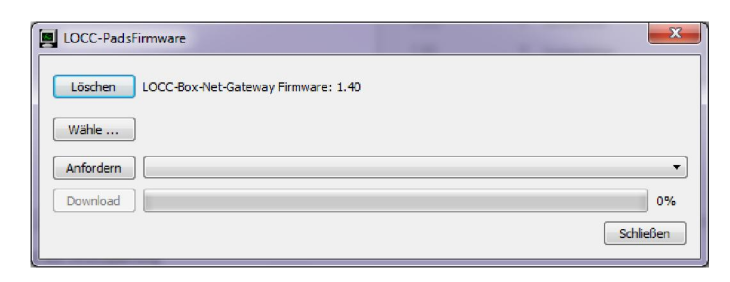
2. LOCC-Pads wird im inaktiven Zustand gestartet. Wählen Sie im Menü **Extra / COM Einstellungen** Ihren Comport aus.

3. Gehen Sie ins Menü **Extra / Firmware Download**. Es erscheint das Feld für die Passworteingabe. Dieses lautet: *"Luetze71384Weinstadt"*. Bitte mit OK bestätigen.



4. Es öffnet sich das Fenster **LOCC-PadsFirmware**. Hier wird Ihnen die momentan verwendete Version angezeigt. Bitte vergleichen Sie diese mit der aktuell herunter geladenen Version. Bei gleichem Versionsstand verlassen Sie bitte das Firmware-Update mit der Schaltfläche "Schließen". Ist die herunter geladene Version höher, dann

wählen Sie “Löschen“. Es erscheint eine Sicherheitsabfrage, welche Sie bitte mit “Ja“ bestätigen. Das Löschen kann bis zu 1 Minute dauern!



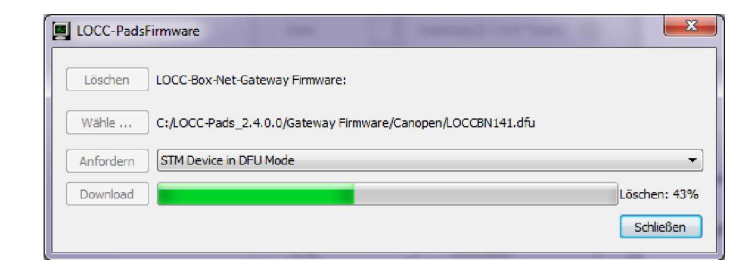


5. Schließen Sie bitte LOCC-Pads, unterbrechen Sie für ca. 5s die Versorgungsspannung und entfernen Sie das USB-Kabel vom Gateway.

6. Nachdem Zuschalten der Versorgungsspannung und dem Verbinden mit dem USB-Kabel wird ggf. das Gateway als neue Hardware erkannt. In diesen Fall lesen Sie bitte Kapitel 4.5 “Neue Hardware installieren“, ansonsten mit Punkt 7 weiter fortfahren.

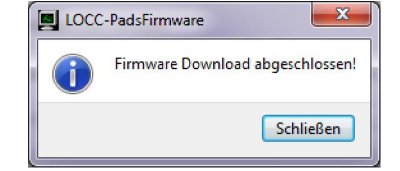
7. Starten Sie LOCC-Pads erneut und öffnen Sie das Fenster LOCC-PadsFirmware (siehe 4.).

8. Drücken Sie die Schalfläche “Wählen“, um die neue Firmware Datei auszuwählen. Format: LOCCBNxxx.dfu
   Diese befindet sich im entpackten LOCC-Pads_x.x.x.x.zip – File im Ordner Gateway Firmware/ CANopen.



9. Betätigen Sie die Schalfläche “Anfordern“, um den Gerätetreiber des verwendeten Microcontroller zu erhalten (STM Device in DFU Mode).

10. Drücken Sie auf “Download“, um mit dem Update zu beginnen. Der Abschluss des Updates wird durch eine weitere Meldung angezeigt.



11. Schließen Sie bitte LOCC-Pads erneut, unterbrechen Sie für ca. 5s die Versorgungsspannung und entfernen Sie das USB-Kabel vom Gateway.

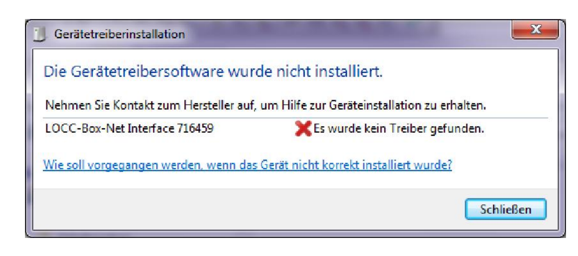
12. Das Gateway ist jetzt auf dem neuesten Stand und kann ohne Einschränkungen verwendet werden.

HINWEIS

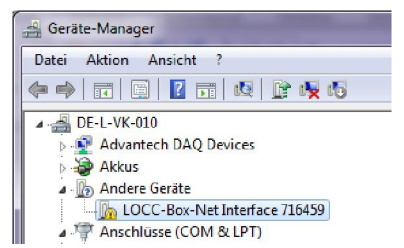
Nachdem Zuschalten der Versorgungsspannung und dem Verbinden mit dem USB-Kabel wird das Gateway als „LOCC-Box-GW-EC 716456“ erkannt, jedoch der Gerätetreiber nicht automatisch installiert. In diesen Fall lesen Sie bitte Kapitel 4.5 “Neue Hardware installieren“.

4.5 Neue Hardware installieren

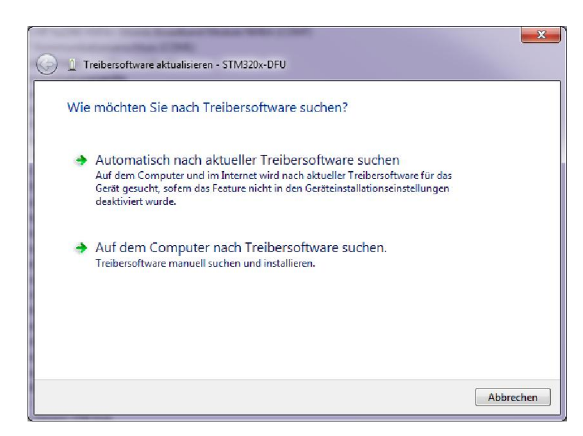
1. Nachdem Zuschalten der Versorgungsspannung wird das Gateway als neue Hardware “STM320x-DFU“ bzw. „LOCC-Box-Net Interface 716459“ erkannt, jedoch wird der Gerätetreiber nicht automatisch installiert.
   Eine manuelle Installation ist erforderlich.



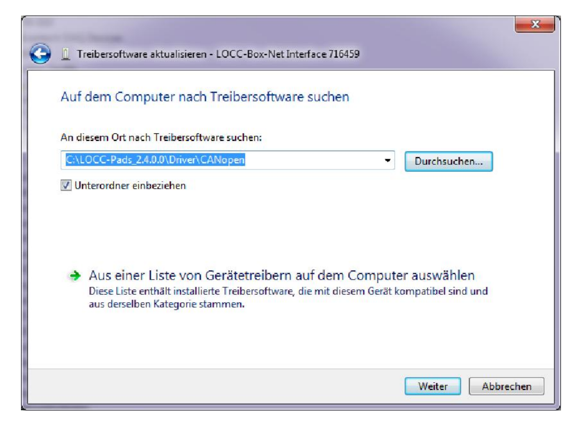
2. Im Geräte-Manager wird die Hardware unter *„Andere Geräte“* angezeigt. Mit einem Rechtsklick auf “ LOCC-Box-Net Interface 716459“ wählen Sie *„Treibersoftware aktualisieren“* aus.



3. Im folgenden Fenster bitte „Auf dem Computer nach Treibersoftware suchen“ klicken.

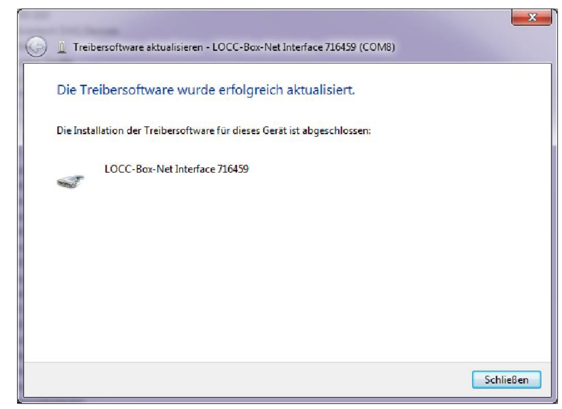


4. Wählen Sie mit *„Durchsuchen“* im entpackten LOCC-Pads_x.x.x.x.zip – File den Ordner Driver/ CANopen aus und bestätigen Sie mit *„Weiter“.* Die folgende Windows-Sicherheitswarnug mit *„Diese Treibersoftware trotzdem installieren“* bestätigen.



5. Bestätigen Sie die erfolgreiche Installation der Treibersoftware mit dem Button *„Schließen“.*

   Die Installation ist beendet.



5 Zubehör

Für das LOCC-Box-Net System steht ein umfangreiches Zubehör zur Verfügung, welches in folgender Tabelle zusammen gefasst ist.

Zubehör	Art.-Nr.	Typ	VPE
Module			
Einspeiseklemme mit Ausbruch für die Kupferschiene zur Stromerhöhung	716421	LOCC-Box-EKL 7-6421	2
Distanzklemme ohne Kontakte	716422	LOCC-Box-DKL 7-6422	2
LOCC-Box Leergehäuse ohne Klemmen	716424	LOCC-Box-DY 7-6424	2
Einspeiseset (Einspeise- und Endklemme)	716425	LOCC-Box-ES 7-6425	1
0V – Sammelklemme	716420	LOCC-Box-SK 7-6420	2
Brückungskamm			
Brückungskamm 8polig, 6A, weiß	716428	LOCC-Box-BKW 7-6428	5
Brückungskamm 8polig, 6A, rot	716429	LOCC-Box-BKR 7-6429	5
Brückungskamm 8polig, 6A, blau	716430	LOCC-Box-BKB 7-6430	5
Brückungskamm 16polig, 6A, weiß	716438	LOCC-Box-BKW 7-6438	5
Brückungskamm 16polig, 6A, rot	716439	LOCC-Box-BKW 7-6439	5
Brückungskamm 16polig, 6A, blau	716440	LOCC-Box-BKW 7-6440	5
Bezeichnungsträger			
Bezeichnungsträger 5x5mm , 200 Stück, weiß	716431	LOCC-Box-BZW 7-6431	1
Bezeichnungsträger 5x5mm, 200 Stück, rot	716432	LOCC-Box-BZR 7-6432	1
Bezeichnungsträger 5x5mm, 200 Stück, blau	716433	LOCC-Box-BZB 7-6433	1
Bezeichnungsträger 5x5mm, 200 Stück, gelb	716434	LOCC-Box-BZG 7-6434	1
Bezeichnungsträger 12x6mm ,160 Stück, weiß	716441	LOCC-Box-BZW 7-6441	1
Bezeichnungsträger 39,3x8mm, weiß	716443	LOCC-Box-BZT 7-6443	20
Abdeckung für 716443, transparent	716444	LOCC-Box-BAD 7-6444	20
A4 Beschriftungsbogen für 716443	716445	LOCC-Box-LEB 7-6445	240
Sonstiges			
Kupferschiene 1m	716426	LOCC-Box-CU 7-6426	1
Abdeckung Kupferschiene 1m	716427	LOCC-Box-AD 7-6427	1

Scope of this document

This gateway may include software licensed by 3rd parties. The following third party intellectual property (IP) notices are provided to comply with the terms of such licenses.

The firmware of the Gateway use the FreeRTOS ^{TM} operating system which is developed under the terms of the GPL. As a special exception to the GPL, the copyright holder of FreeRTOS gives the permission to link FreeRTOS with independent modules that communicate with FreeRTOS solely through the FreeRTOS API interface, regardless of the license terms of these independent modules, and to copy and distribute the resulting combined work without being obliged to provide the source code of these proprietary modules. See the licensing section of http://www.freeRTOS.org for details.

1) Windows, Windows 2000, Windows XP, Windows Vista und Windows7 sind eingetragene Warenzeichen der Microsoft Corporation.